

.vscode/settings.json

```
{
  "files.watcherExclude": {
    "**/target": true
  }
}
```

backend/connection/atendimento\_db.js

```
import { connect } from "../db_connect.js";
```

```
const btnFinaliza = document.querySelector("#btn-finaliza-consulta");
```

```
function ajustaIndicadores() {
  let indicadores = "";

  let baixoRisco = [];
  baixoRisco[0] = localStorage.getItem('brIdade');
  baixoRisco[1] = localStorage.getItem('brSexMasc');
  baixoRisco[2] = localStorage.getItem('brTabag');
  baixoRisco[3] = localStorage.getItem('brHipert');
  baixoRisco[4] = localStorage.getItem('brObs');
  baixoRisco[5] = localStorage.getItem('brSedent');
  baixoRisco[6] = localStorage.getItem('brHistoFam');
  baixoRisco[7] = localStorage.getItem('brHistoDoenca');
  baixoRisco[8] = localStorage.getItem('brManifest');
  baixoRisco[9] = localStorage.getItem('brDiagnDislip');
  baixoRisco[10] = localStorage.getItem('brDiagnPolic');

  let listBaixo = "";
  for (let i = 0; i < baixoRisco.length; i++) {
    if (baixoRisco[i] != 'vazio') {
      listBaixo += baixoRisco[i] + "\n";
    }
  }

  if (listBaixo != "") {
    indicadores = "- Baixo Risco:\n" + listBaixo;
  }

  let altoRisco = [];
  altoRisco[0] = localStorage.getItem('arAcidVasc');
  altoRisco[1] = localStorage.getItem('arInfartAgud');
  altoRisco[2] = localStorage.getItem('arLesPerif');
  altoRisco[3] = localStorage.getItem('arAtaqlsqu');
  altoRisco[4] = localStorage.getItem('arHiperVentri');
  altoRisco[5] = localStorage.getItem('arNefrop');
  altoRisco[6] = localStorage.getItem('arRetin');
  altoRisco[7] = localStorage.getItem('arAneur');
  altoRisco[8] = localStorage.getItem('arEsten');
  altoRisco[9] = localStorage.getItem('arDiabet');
  altoRisco[10] = localStorage.getItem('arDoenVasc');
  altoRisco[11] = localStorage.getItem('arInsuf');
  altoRisco[12] = localStorage.getItem('arAngina');
  altoRisco[13] = localStorage.getItem('arDoencRenal');
```

```

let listAlto = "";
for (let i = 0; i < altoRisco.length; i++) {
    if (altoRisco[i] != 'vazio') {
        listAlto += altoRisco[i] + "\n";
    }
}

if (listAlto != "") {
    indicadores += "- Alto Risco:\n" + listAlto;
}

return indicadores;
}

function generalList(nome, dig) {
    let lista = "";
    let tot = localStorage.getItem(`qtd-${nome}`);
    for (let i = 0; i < dig; i++) {
        let item = `${nome}-${i}`;
        let check = localStorage.getItem(item);
        // console.log(check);
        if (check != null)
            lista += check + ".\n";
    }
    if (lista != "") {
        lista = 'Diagnósticos/Resultados de enfermagem: \n' + lista + 'Intervenções de Enfermagem: \n';
        for (let i = dig; i < tot; i++) {
            let item = `${nome}-${i}`;
            let check = localStorage.getItem(item);
            // console.log(check);
            if (check != null)
                lista += check + ".\n";
        }
    }
    return lista;
}

function ajustaListPsicobio() {
    let lista_final = "";

    //OXIGENAÇÃO
    let oxi = generalList('oxi', 2);
    if (oxi != "") {
        lista_final += "- Oxigenação:\n" + oxi + "\n";
    }

    //NUTRIÇÃO
    let nut = generalList('nut', 17);
    if (nut != "") {
        lista_final += "- Nutrição:\n" + nut + "\n";
    }

    //SONO E REPOUSO
    let son = generalList('son', 3);
    if (son != "") {
        lista_final += "- Sono e Repouso:\n" + son + "\n";
    }
}

```

```

}

//ATIVIDADE FÍSICA
let fis = generalList('fis', 5);
if (fis != "") {
    lista_final += "- Atividade Física:\n" + fis + "\n";
}

//REG. VASCULAR
let vas = generalList('vas', 15);
if (vas != "") {
    lista_final += "- Regulação Vascular:\n" + vas + "\n";
}

//REG. HORMONAL
let hor = generalList('hor', 1);
if (hor != "") {
    lista_final += "- Regulação Hormonal:\n" + hor + "\n";
}

//REG. NEUROLÓGICA
let neu = generalList('neu', 3);
if (neu != "") {
    lista_final += "- Regulação Neurológica:\n" + neu + "\n";
}

//ÓRGÃOS DO SENTIDO
let sen = generalList('sen', 2);
if (sen != "") {
    lista_final += "- Órgãos do Sentido:\n" + sen + "\n";
}

//TERAPÊUTICA
let ter = generalList('ter', 4);
if (ter != "") {
    lista_final += "- Terapêutica:\n" + ter + "\n";
}

console.log(lista_final);
return lista_final;
}

function ajustaListPsicossoc() {
    let lista_final = "";

    //AMOR E ACEITAÇÃO
    let amo = generalList('amo', 1);
    if (amo != "") {
        lista_final += "- Amor e Aceitação:\n" + amo + "\n";
    }

    //LIBERDADE E PARTICIPAÇÃO
    let lib = generalList('lib', 2);
    if (lib != "") {
        lista_final += "- Liberdade e Aceitação:\n" + lib + "\n";
    }
}

```

```

//EDUCAÇÃO PARA SAÚDE E APRENDIZAGEM
let edu = generalList('edu', 5);
if (edu != '') {
    lista_final += "- Educação para Saúde e Aprendizagem:\n" + edu + "\n";
}

console.log(lista_final);
return lista_final;
}

function ajustaListPsicoesp() {
    let lista_final = "";

    //RELIGIOSIDADE E ESPIRITUALIDADE
    let reg = generalList('reg', 1);
    if (reg != '') {
        lista_final = "- Religiosidade e Espiritualidade:\n" + reg + "\n";
    }

    console.log(lista_final);
    return lista_final;
}

export async function saveConsulta() {
    let date = new Date();
    let cur_date = date.getFullYear() + '-' + (date.getMonth() + 1) + '-' + date.getDate() + ' ' +
    date.getHours() + ':' + date.getMinutes() + ':' + date.getSeconds();

    let idUserSus = localStorage.getItem('idUserSus');
    let pas = localStorage.getItem('pas');
    let pad = localStorage.getItem('pad');
    let clas_ha = localStorage.getItem('clas_ha');
    let imc = localStorage.getItem('imc');
    let clas_imc = localStorage.getItem('clas_imc');
    let circ_abdo = localStorage.getItem('circAbdomCm');
    let clas_abdo = localStorage.getItem('clas_abdo');
    let cint_quad = localStorage.getItem('cint_quad');
    let clas_cint_quad = localStorage.getItem('clas_cint_quad');
    let torn_brac = localStorage.getItem('torn_brac');
    let clas_torn_braq = localStorage.getItem('clas_torn_braq');

    //INDICADORES
    let indicadores = ajustaIndicadores();

    let clas_estrat = localStorage.getItem('clas_estrat');
    let list_psicobio = ajustaListPsicobio();
    let list_psicosoc = ajustaListPsicossoc();
    let list_psicoesp = ajustaListPsicoesp();

    let hemogHemac = localStorage.getItem('hemogHemac');
    let hemogHemogl = localStorage.getItem('hemogHemogl');
    let hemogHemat = localStorage.getItem('hemogHemat');
    let hemogHeuc = localStorage.getItem('hemogHeuc');
    let hemogHinf = localStorage.getItem('hemogHinf');
    let hemogHlaq = localStorage.getItem('hemogHlaq');
    let hemogHlic = localStorage.getItem('hemogHlic');

```

```

let hemogColes = localStorage.getItem('hemogColes');
let hemogHdl = localStorage.getItem('hemogHdl');
let hemogTrig = localStorage.getItem('hemogTrig');
let hemogHemogGlic = localStorage.getItem('hemogHemogGlic');
let hemogCreat = localStorage.getItem('hemogCreat');
let hemogPotass = localStorage.getItem('hemogPotass');
let hemogEqu = localStorage.getItem('hemogEqu');
let hemogMicroa = localStorage.getItem('hemogMicroa');
let hemogAlbum = localStorage.getItem('hemogAlbum');
let hemogAcid = localStorage.getItem('hemogAcid');
let hemogUreia = localStorage.getItem('hemogUreia');
let hemogTgo = localStorage.getItem('hemogTgo');
let hemogTgp = localStorage.getItem('hemogTgp');

// console.log(ajustaIndicadores());
const conn = await connect();
let [results] = await conn.query(
  'INSERT INTO consulta (data_cons, fk_id_usersus, pas, clas_ha, imc, clas_imc, clas_abdo,
  clas_cint_quad, clas_torn_brac, indicadores, clas_estrat, list_psicobio, list_psicosoc, list_psicoesp, pad,
  circ_abdo, cint_quad, torn_brac) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)',
  [cur_date, idUserSus, pas, clas_ha, imc, clas_imc, clas_abdo, clas_cint_quad, clas_torn_brac,
  indicadores, clas_estrat, list_psicobio, list_psicosoc, list_psicoesp, pad, circ_abdo, cint_quad,
  torn_brac],
  function (err) {
    if (err) throw err;
  }
);

let [results2] = await conn.query(
  'INSERT INTO hemograma (fk_id_usersus, fk_data_cons, hemacias, hemoglobina, hematocrito,
  leucocitos, linfocitos, plaquetas, glicemia_jejum, colesterol_total, hdl, triglicerideos,
  hemoglobina_glicada, creatina, potassio_plasmatico, equ, microalbumina_urina, albumina,
  acido_urico, ureia, tgo, tgp) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)',
  [idUserSus, cur_date, hemogHemac, hemogHemogl, hemogHemat, hemogHeuc, hemogHinf,
  hemogHlaq, hemogHlic, hemogColes, hemogHdl, hemogTrig, hemogHemogGlic, hemogCreat,
  hemogPotass, hemogEqu, hemogMicroa, hemogAlbum, hemogAcid, hemogUreia, hemogTgo,
  hemogTgp],
  function (err) {
    if (err) throw err;
  }
);
conn.end();
return results.affectedRows;
}

```

```

btnFinaliza.addEventListener('click', saveConsulta);

```

```

backend/connection/consulta_db.js

```

```

import { connect } from "../db_connect.js";

```

```

//BUSCAR ALTO RISCO

```

```

export async function searchAltoRisco(idUserSus){
  const conn = await connect();
  let [row] = await conn.query("SELECT clas_estrat FROM consulta WHERE fk_id_usersus = ? ORDER
  BY data_cons DESC LIMIT 1", [idUserSus]);
}

```

```

    await conn.end();
    // console.log(row[0]);
    if (row[0] === undefined){
        // console.log("NÃO TEM");
        return 'indefinido';
    }
    else {
        // console.log(row[0].clas_estrat);
        return row[0].clas_estrat;
    }
}

//BUSCA TODOS ALTO RISCO
export async function searchAllAltoRisco(){
    const conn = await connect();
    let [rows] = await conn.query(`
        SELECT cart_sus, cpf, nome, sexo, data_nasc FROM user_sus u
        JOIN consulta c ON u.id_usersus = c.fk_id_usersus
        WHERE c.clas_estrat LIKE "ALTO RISCO"
        AND c.data_cons = (
            SELECT MAX(data_cons) FROM consulta
            WHERE fk_id_usersus = u.id_usersus
        )`
    );
    await conn.end();
    return rows;
}

//BUSCAR ULTIMA CONSULTA
export async function searchUltimaConsulta(idUserSus){
    const conn = await connect();
    let [row] = await conn.query("SELECT * FROM consulta WHERE fk_id_usersus = ? ORDER BY
data_cons DESC LIMIT 1", [idUserSus]);
    await conn.end();
    // console.log(row[0]);
    if (row[0] === undefined){
        // console.log("NÃO TEM");
        return 'indefinido';
    }
    else {
        // console.log(row[0]);
        return row[0];
    }
}
}

```

backend/connection/db\_connect.js

```

export async function connect(){
    // console.log("Tentando conectar!");
    const mysql = require("mysql2/promise");
    const connection = await
mysql.createConnection("mysql://root:12345@localhost:3306/assist_enferm");
    // console.log("Conectou ao MySQL!");
    return connection;
}

```

backend/connection/equipe\_db.js

```
import { connect } from "../db_connect.js";

//LOGIN
export async function selectEquipe(login, senha){
  const conn = await connect();
  let [rows] = await conn.query('SELECT * FROM usuario WHERE login=? AND senha=?', [login, senha]);
  await conn.end();
  return rows;
}

//CADASTRO
export async function equipeExists(login){
  const conn = await connect();
  let [rows] = await conn.query('SELECT * FROM usuario WHERE login=?', [login]);
  await conn.end();
  return rows;
}

//ALTERAR
export async function loadEquipe(idEquipe){
  const conn = await connect();
  let [row] = await conn.query('SELECT * FROM usuario WHERE id=?', [idEquipe]);
  await conn.end();
  return row;
}

//BUSCAR
export async function searchEquipes(login){
  const conn = await connect();
  login = '%' + login + '%';
  let [rows] = await conn.query('SELECT * FROM usuario WHERE login LIKE ?', [login]);
  await conn.end();
  return rows;
}

export async function insertEquipe(login, senha, membros){
  const conn = await connect();
  let [results] = await conn.query(
    'INSERT INTO usuario (login, senha, membros) VALUES (?, ?, ?)',
    [login, senha, membros],
    function(err){
      if(err) throw err;
    }
  );
  conn.end();
  return results.affectedRows;
}

export async function updateEquipes(idEquipe, login, senha, membros){
  const conn = await connect();
  let [results] = await conn.query(
    'UPDATE usuario SET login=?, senha=?, membros=? WHERE id=?',
    [login, senha, membros, idEquipe],
    function(err){
```

```

        if(err) throw err;
    }
    );
    conn.end();
    return results.affectedRows;
}

```

backend/connection/userSus\_db.js

```
import { connect } from "../db_connect.js";
```

```
//CADASTRO
```

```
export async function userSusExists(cartSus, cpf) {
    const conn = await connect();
    let [rows] = await conn.query('SELECT * FROM user_sus WHERE cart_sus=? OR cpf=?', [cartSus, cpf]);
    await conn.end();
    return rows;
}

```

```
export async function lastUserSus() {
    const conn = await connect();
    let [rows] = await conn.query('SELECT MAX(id_usersus) AS lastId FROM user_sus');
    await conn.end();
    return rows;
}

```

```
//ALTERAR
```

```
export async function loadUserSus(idUserSus) {
    const conn = await connect();
    let [row] = await conn.query('SELECT * FROM user_sus WHERE id_usersus=?', [idUserSus]);
    await conn.end();
    return row;
}

```

```
//BUSCAR
```

```
export async function searchUserSus(user, idEquipe) {
    const conn = await connect();
    user = '%' + user + '%';
    let [rows] = await conn.query(
        "SELECT * FROM user_sus WHERE fk_id_equipe=? AND (cart_sus LIKE ? OR cpf LIKE ? OR nome LIKE ?)",
        [idUserSus, user, user, user]);
    await conn.end();
    return rows;
}

```

```
export async function insertUserSus(cartSus, cpf, nome, dataNasc, sexo, idEquipe) {
    const conn = await connect();
    let [results] = await conn.query(
        'INSERT INTO user_sus (cart_sus, cpf, nome, data_nasc, sexo, fk_id_equipe) VALUES (
        ?, ?, ?, ?, ?, ?)',
        [cartSus, cpf, nome, dataNasc, sexo, idEquipe],
        function (err) {
            if (err) throw err;
        }
    );
}

```



```

    );
    conn.end();
    return results.affectedRows;
}

export async function todos_users() {
    const conn = await connect();
    let [rows] = await conn.query('SELECT * FROM user_sus');
    await conn.end();
    return rows;
}

// export async function updateUserSus(cartSus, cpf, nome, dataNasc, sexo, idEquipe){
//     const conn = await connect();
//     let [results] = await conn.query(
//         'UPDATE user_sus SET cpf=?, nome=?, data_nasc=?, sexo=?, fk_id_equipe=? WHERE
//         cart_sus=?',
//         [cpf, nome, dataNasc, sexo, idEquipe, cartSus],
//         function(err){
//             if(err) throw err;
//         }
//     );
//     conn.end();
//     return results.affectedRows;
// }

```

backend/equipe\_js/busca.js

```
import { searchEquipes } from "../connection/equipe_db.js";
```

```
const btnBusca = document.querySelector("#buscar");
const retornoBuscaEquipe = document.querySelector('#retorno-busca-equipe');
```

```
const isEmpty = str => !str.trim().length;
```

```

async function buscar() {
    let login = document.querySelector("#login").value;
    let result = document.querySelector("#result");

    result.innerHTML = "";

    if (isEmpty(login)) {
        retornoBuscaEquipe.innerHTML = "ERRO -> Nenhum valor inserido na busca!";
        retornoBuscaEquipe.style.color = 'red';
        setTimeout(function () {
            retornoBuscaEquipe.innerHTML = "";
        }, 5000);
    }
    else {
        let rows = await searchEquipes(login);
        // console.log(rows.length);

        if (rows.length > 0) {
            for (let i = 0; i < rows.length; i++) {
                result.innerHTML += `<button class="equipe" onclick="selecionarEquipe(this)"
                value="${rows[i].id}">Login: ${rows[i].login} / Membros: ${rows[i].membros}</button><br>`;
            }
        }
    }
}

```

```

    }
  } else {
    retornoBuscaEquipe.innerHTML = "Não foi encontrada nenhuma equipe com este login!!";
    retornoBuscaEquipe.style.color = 'green';
    setTimeout(function () {
      retornoBuscaEquipe.innerHTML = "";
    }, 3000);
  }
}
}

```

```

btnBusca.addEventListener("click", buscar);

```

```

import { insertEquipe, equipeExists, selectEquipe } from "../connection/equipe_db.js";

```

```

const btnCadast = document.querySelector("#cadastrar");

```

```

const retornoCadEquipe = document.querySelector("#retorno-cad-equipe");

```

```

async function cadastrar(){
  let login = document.querySelector("#login").value;
  let senha = document.querySelector("#senha").value;
  let membros = document.querySelector("#membros").value;

  let rows = await equipeExists(login);

  if(rows.length > 0){
    retornoCadEquipe.innerHTML = "Equipe já cadastrada no sistema!!";
    //window.alert('Equipe já cadastrada no sistema!!');
    setTimeout(function(){
      retornoCadEquipe.innerHTML = "";
    }, 3000);
  } else {
    let result = await insertEquipe(login, senha, membros);
    if(result > 0){
      /*Busca id da ultima equipe cadastrada*/
      let idExt = await selectEquipe(login, senha);
      localStorage.setItem('id_equipe', idExt[0].id);
      if(idExt[0].id == 1){
        localStorage.setItem('ehAdmin', 'ehAdmin');
      }else{
        localStorage.setItem('ehAdmin', 'naoEhAdmin');
      }
      console.log('ID: ', idExt[0].id);
      retornoCadEquipe.innerHTML = "Equipe cadastrada com sucesso.";
      retornoCadEquipe.style.color = 'green';
      setTimeout(function(){
        retornoCadEquipe.innerHTML = "";
        window.location = "./adm_area.html";
      }, 2000);
    }
  }
  else {
    retornoCadEquipe.innerHTML = "ERRO!\nNão foi possível cadastrar a equipe!!";
    retornoCadEquipe.style.color = 'red';
  }
}

```

```

        setTimeout(function(){
            retornoCadEquipe.innerHTML = "";
        }, 3000);
    }
}
}
btnCadast.addEventListener("click", cadastrar);

backend/equipe_js/cadastro.js

import { insertEquipe, equipeExists, selectEquipe } from "../connection/equipe_db.js";

const btnCadast = document.querySelector("#cadastrar");
const retornoCadEquipe = document.querySelector("#retorno-cad-equipe");

async function cadastrar(){
    let login = document.querySelector("#login").value;
    let senha = document.querySelector("#senha").value;
    let membros = document.querySelector("#membros").value;

    let rows = await equipeExists(login);

    if(rows.length > 0){
        retornoCadEquipe.innerHTML = "Equipe já cadastrada no sistema!!";
        //window.alert('Equipe já cadastrada no sistema!!');
        setTimeout(function(){
            retornoCadEquipe.innerHTML = "";
        }, 3000);
    } else {
        let result = await insertEquipe(login, senha, membros);
        if(result > 0){
            /*Busca id da ultima equipe cadastrada*/
            let idExt = await selectEquipe(login, senha);
            localStorage.setItem('id_equipe', idExt[0].id);
            if(idExt[0].id == 1){
                localStorage.setItem('ehAdmin', 'ehAdmin');
            } else {
                localStorage.setItem('ehAdmin', 'naoEhAdmin');
            }
            console.log('ID: ', idExt[0].id);
            retornoCadEquipe.innerHTML = "Equipe cadastrada com sucesso.";
            retornoCadEquipe.style.color = 'green';
            setTimeout(function(){
                retornoCadEquipe.innerHTML = "";
                window.location = "../adm_area.html";
            }, 2000);
        }
    }
    else {
        retornoCadEquipe.innerHTML = "ERRO!\nNão foi possível cadastrar a equipe!!";
        retornoCadEquipe.style.color = 'red';
        setTimeout(function(){
            retornoCadEquipe.innerHTML = "";
        }, 3000);
    }
}

```

```

    }
  }
}
btnCadast.addEventListener("click", cadastrar);

```

backend/equipe\_js/login.js

```

import { selectEquipe } from "../connection/equipe_db.js";

const btnLogin = document.querySelector("#logar");

async function logar(){
  // console.log("COMEÇOU!");

  let login = document.querySelector("#login").value;
  let senha = document.querySelector("#senha").value;

  let retornoLogin = document.querySelector("#acesso-negado");

  // let sql = `SELECT * FROM usuario WHERE login='${login}' AND senha='${senha}'`;
  // console.log(sql);

  let rows = await selectEquipe(login, senha);
  // console.log(rows);

  if(rows.length > 0){
    // console.log("Usuário encontrado!");
    localStorage.setItem('id_equipe', rows[0].id);
    if(rows[0].id == 1){
      localStorage.setItem('ehAdmin', 'ehAdmin');
    }else{
      localStorage.setItem('ehAdmin', 'naoEhAdmin');
    }
    window.location = "../pages/adm_area.html";
  } else {
    // console.log("Acesso NEGADO!!");
    retornoLogin.innerHTML = 'Falha ao acessar o sistema!\nUsuário ou Senha incorreto(s)!';
    retornoLogin.style.color = 'red';
    setTimeout(function(){
      retornoLogin.innerHTML = "";
    }, 5000);
  }
}

btnLogin.addEventListener("click", logar);

```

backend/equipe\_js/seleciona.js

```

function selecionarEquipe(equipe){
  let idEquipe = equipe.value;
  // console.log(idEquipe);
  localStorage.setItem('id_equipe', idEquipe);
  window.location = "../pages/alt_equipe.html";
}

```

backend/equipe\_js/update.js

```
import { updateEquipes, loadEquipe } from "../connection/equipe_db.js";

const btnAlterar = document.querySelector("#alterar");
const retornoUpEquipe = document.querySelector("#retorno-up-equipe");
const idEquipe = localStorage.getItem('id_equipe');

async function alterarEquipe(){
  console.log(idEquipe);
  let login = document.querySelector("#login");
  let senha = document.querySelector("#senha");
  let membros = document.querySelector("#membros");

  let result = await updateEquipes(idEquipe, login.value, senha.value, membros.value);
  if(result > 0){
    retornoUpEquipe.innerHTML = 'Equipe atualizada com sucesso.';
    retornoUpEquipe.style.color = 'green';
    setTimeout(function(){
      window.location = "../pages/adm_area.html";
    }, 3000);
  }
  else {
    retornoUpEquipe.innerHTML = 'ERRO!\nNão foi possível atualizar a equipe!!';
    retornoUpEquipe.style.color = 'red';
    setTimeout(function(){
      retornoUpEquipe.innerHTML = "";
    }, 3000);
  }
}

window.onload = async function(){
  let result = await loadEquipe(idEquipe);

  let login = document.querySelector("#login");
  let senha = document.querySelector("#senha");
  let membros = document.querySelector("#membros");

  login.value = result[0].login;
  senha.value = result[0].senha;
  membros.value = result[0].membros;
}

btnAlterar.addEventListener("click", alterarEquipe);
```

backend/relatorio/relAltoRisco.js

```
import { searchAllAltoRisco } from "../connection/consulta_db.js";

async function printAltoRisco() {
  let result = document.querySelector("#retorno-select-2");
  let table = `
    <table>
      <tr>
        <th> Cartão do SUS </th>
```

```

        <th> CPF </th>
        <th> Nome </th>
        <th> Data de Nasc. </th>
        <th> Sexo </th>
    </tr>
    `;
    let rows = await searchAllAltoRisco();
    // console.log(rows);
    // console.log(rows.length);

    for (let i = 0; i < rows.length; i++) {
        let data_nasc = rows[i].data_nasc.getDate() + '/' + (rows[i].data_nasc.getMonth() + 1) + '/' +
rows[i].data_nasc.getFullYear();
        table += `
            <tr>
                <td> ${rows[i].cart_sus} </td>
                <td> ${rows[i].cpf} </td>
                <td> ${rows[i].nome} </td>
                <td> ${data_nasc} </td>
                <td> ${rows[i].sexo} </td>
            </tr>
        `;
    }

    table += `
        </table>
    `;

    result.innerHTML = table;
}

window.addEventListener('load', printAltoRisco);

```

backend/relatorio/relConsulta.js

```

import { searchUltimaConsulta } from "../connection/consulta_db.js";
import { loadUserSus } from "../connection/userSus_db.js";

```

```

async function printUltimaConsulta() {
    const idUser = localStorage.getItem('idUserSus');

    let result = document.querySelector(".retorno-select-3");

    let row = await loadUserSus(idUser);

    let data_nasc = row[0].data_nasc.getDate() + '/' + (row[0].data_nasc.getMonth() + 1) + '/' +
row[0].data_nasc.getFullYear();

    let basic_info = `
        <div class="info">
            <h3>Dados do Usuário(a)</h3>
            <div class="infos">
                <div>
                    <p><strong>Nome:</strong> ${row[0].nome}</p>
                    <p><strong>Cartão do SUS:</strong> ${row[0].cart_sus}</p>

```

```

        <p><strong>CPF:</strong> ${row[0].cpf}</p>
    </div>
    <div>
        <p><strong>Data de Nascimento:</strong> ${data_nasc}</p>
        <p><strong>Sexo:</strong> ${row[0].sexo}</p>
    </div>
</div>
</div>
`;

row = await searchUltimaConsulta(idUser);
console.log(row);
let data_consulta = row.data_cons.getDate() + '/' + (row.data_cons.getMonth() + 1) + '/' +
row.data_cons.getFullYear();

let info_consulta = `
    <div class="info">
        <h3>Exame físico/avaliação clínica</h3>
        <p>Data da consulta: ${data_consulta}</p>
        <div class="infos">
            <div class="exames">
                <p><strong>PAS:</strong> ${row.pas} / <strong>PAD:</strong> ${row.pad}</p>
                <p><strong>Classificação da HA:</strong> ${row.clas_ha}</p>
                <p><strong>IMC:</strong> ${row.imc.toFixed(2)}</p>
                <p><strong>Classificação do IMC:</strong> ${row.clas_imc}</p>
            </div>
            <div>
                <p><strong> Classificação Circunferência abdominal:</strong> ${row.clas_abdo}</p>
                <p><strong> Circunferência abdominal:</strong> ${row.circ_abdo}</p>
                <p><strong> Classificação Cintura-Quadril:</strong> ${row.clas_cint_quad}</p>
                <p><strong> Cintura-Quadril:</strong> ${row.cint_quad.toFixed(2)}</p>
                <p><strong> Classificação Índice Tornozelo-Braquial:</strong> ${row.clas_torn_brac}</p>
                <p><strong> Índice Tornozelo-Braquial:</strong> ${row.torn_brac.toFixed(2)}</p>
            </div>
        </div>
    </div>
`;
let indi = row.indicadores.replaceAll("\n", '<br>')
let estrat = `
    <div class="info html2pdf__page-break">
        <h3>Estratificação do Risco Cardiovascular</h3>
        <p><strong>Classificação da Estratificação:</strong> ${row.clas_estrat}</p>
        <p>${indi}</p>
    </div>
`;
let nec_psBio = "";
if (row.list_psicobio != "") {
    let Psicobio = row.list_psicobio.replaceAll("\n", '<br>')
    nec_psBio = `
        <div class="info html2pdf__page-break">
            <h3>Necessidades Psicobiológicas</h3>
            <p>${Psicobio}</p>
        </div>
    `;
}
let nec_psSoc = "";

```

```

if (row.list_psicosoc) {
  let Psicosoc = row.list_psicosoc.replaceAll('\n', '<br>')
  nec_psSoc = `
<div class="info html2pdf__page-break">
  <h3>Necessidades Psicosociais</h3>
  <p>${Psicosoc}</p>
</div>
`;
}
let nec_psEsp = "";
if (row.list_psicoesp) {
  let Psicoesp = row.list_psicoesp.replaceAll('\n', '<br>')
  nec_psEsp = `
<div class="info html2pdf__page-break">
  <h3>Necessidades Psicoespirituais</h3>
  <p>${Psicoesp}</p>
</div>
`;
}
result.innerHTML = basic_info + info_consulta + estrat + nec_psBio + nec_psSoc + nec_psEsp;
}

```

```

window.addEventListener('load', printUltimaConsulta);

```

```

backend/relatorio/relTodosUsers.js

```

```

import { todos_users } from "../connection/userSus_db.js";

```

```

async function printTodosUsers() {
  let result = document.querySelector(".retorno-select-2");
  let table = `
<table>
  <tr>
    <th> Cartão do SUS </th>
    <th> CPF </th>
    <th> Nome </th>
    <th> Data de Nasc. </th>
    <th> Sexo </th>
  </tr>
  `;
  let rows = await todos_users();
  // console.log(rows);
  // console.log(rows.length);

  for (let i = 0; i < rows.length; i++) {
    let data_nasc = rows[i].data_nasc.getDate() + '/' + (rows[i].data_nasc.getMonth() + 1) + '/' +
rows[i].data_nasc.getFullYear();
    table += `
      <tr>
        <td> ${rows[i].cart_sus} </td>
        <td> ${rows[i].cpf} </td>
        <td> ${rows[i].nome} </td>
        <td> ${data_nasc} </td>
        <td> ${rows[i].sexo} </td>
      </tr>
    `;
  }
}

```



```

    }

    table += `
        </table>
    `;

    result.innerHTML = table;
}

window.addEventListener('load', printTodosUsers);

backend/relatorio/relatorios.js

const btnLastCadast = document.querySelector("#btn-print-maior-risco");
const btnBusca = document.querySelector("#btn-print-ult-adic");

async function chooseSearch() {
    let option = document.querySelector(".form-select-pdf").value;

    console.log(option);

    if (option == 1) {
        console.log("Todos usuários do SUS com Alto Risco.");
        window.location = "./relatorioAltoRisco.html";
    }
    else if (option == 2) {
        console.log("Último usuário do SUS cadastrado.");
        window.location = "./relatorioTodosUsers.html";
    }
}

async function buscaTodos() {
    let user = document.querySelector(".busca-usuarios").value;
    let rows = await searchUserSus(user, idEquipe);
    let dataAtual = parseInt(new Date().toLocaleString().substring(6, 10));
    for (let i = 0; i < rows.length; i++) {
        dataInformada = parseInt((select).toLocaleString().substring(0, 4));

        table += `
            <tr>
                <td> ${rows[i].nome} </td>
                <td> ${dataAtual - dataInformada} </td>
                <td> ${rows[i].sexo} </td>
            </tr>
        `;
    }

    table += `
    </table>
    `;
    //botao (consulta 1)
    //botao (consulta 2)

}

btnLastCadast.addEventListener("click", chooseSearch);
btnBusca.addEventListener("click", buscaTodos);

```

backend/user\_sus/busca.js

```
import { searchUserSus } from "../connection/userSus_db.js";
import { searchAltoRisco } from "../connection/consulta_db.js";

const btnBusca = document.querySelector("#buscar");
const idEquipe = localStorage.getItem('id_equipe');
const retornoBuscaUser = document.querySelector('#retorno-busca-user');

const isEmpty = str => !str.trim().length;

async function buscar() {
  let user_sus = document.querySelector("#user_sus").value.trim();
  let result = document.querySelector("#result");

  result.innerHTML = "";

  if (isEmpty(user_sus)) {
    retornoBuscaUser.innerHTML = "ERRO -> Nenhum valor inserido na busca!";
    retornoBuscaUser.style.color = 'red';
    setTimeout(function () {
      retornoBuscaUser.innerHTML = "";
    }, 5000);
  }
  else {
    let rows = await searchUserSus(user_sus, idEquipe);
    // console.log(rows);

    if (rows.length > 0) {
      for (let i = 0; i < rows.length; i++) {
        if (await searchAltoRisco(rows[i].id_usersus) === 'ALTO RISCO') {
          result.innerHTML += `<button class="equipe" onclick="selecionarUserSus(this)"
value="${rows[i].id_usersus}">
  <span class="flag-risco"></span>
  Cartão SUS: ${rows[i].cart_sus}<br>
  Nome: ${rows[i].nome} /
  CPF: ${rows[i].cpf}</button><br>`;
        }
        else {
          result.innerHTML += `<button class="equipe" onclick="selecionarUserSus(this)"
value="${rows[i].id_usersus}">
  Cartão SUS: ${rows[i].cart_sus}<br>
  Nome: ${rows[i].nome} /
  CPF: ${rows[i].cpf}</button><br>`;
        }
      }
    }
    else {
      retornoBuscaUser.innerHTML = 'Nenhum usuário(a) do SUS encontrado(a)!!!';
      retornoBuscaUser.style.color = 'green';
      setTimeout(function () {
        retornoBuscaUser.innerHTML = "";
      }, 3000);
    }
  }
}

btnBusca.addEventListener("click", buscar);
```

backend/user\_sus/cadastro.js

```
import { insertUserSus, userSusExists, lastUserSus } from "../connection/userSus_db.js";

const btnCadast = document.querySelector("#cadastrar");
const idEquipe = localStorage.getItem('id_equipe');
const retornoCadUserSucess = document.querySelector('#retorno-cad-user-sucess');

async function cadastrar(){
    let cartSus = document.querySelector("#cartSus").value;
    let cpf = document.querySelector("#cpf").value;
    let nome = document.querySelector("#nome").value;
    let dataNasc = document.querySelector("#dataNasc").value;
    let sexo = document.querySelector('input[name="sexo"]:checked').value;

    let rows = await userSusExists(cartSus, cpf);

    if(rows.length > 0){
        //window.alert('Usuário(a) do SUS já cadastrado(a) no sistema!!');
        retornoCadUserSucess.innerHTML = 'Usuário(a) do SUS já cadastrado(a) no sistema!!!';
    } else {
        let result = await insertUserSus(cartSus, cpf, nome, dataNasc, sexo, idEquipe);
        if(result > 0){
            retornoCadUserSucess.innerHTML = 'Usuário(a) cadastrado(a) com sucesso.';
            //window.alert('Usuário(a) cadastrado(a) com sucesso.');
```

```
            let last = await lastUserSus();
            // console.log(last[0].lastId);
            localStorage.setItem('idUserSus', last[0].lastId);
            setTimeout(function(){
                window.location = "../pages/menu2.html";

            }, 2000);
        }
        else
            retornoCadUserSucess.innerHTML = 'ERRO!\nNão foi possível cadastrar o(a) Usuário(a)!!!';
            //window.alert('ERRO!\nNão foi possível cadastrar o(a) Usuário(a)!!!');
    }
}

btnCadast.addEventListener("click", cadastrar);
```

backend/user\_sus/loadDataUser.js

```
import { loadUserSus } from '../connection/userSus_db.js';

const idUserSus = localStorage.getItem('idUserSus');

window.onload = async function () {
    let result = await loadUserSus(idUserSus);

    //console.log(result[0].data_nasc);

    let data_nasc = result[0].data_nasc.getFullYear() + '-' + (result[0].data_nasc.getMonth() + 1) + '-' +
    result[0].data_nasc.getDate();

    // console.log(result[0].sexo);
    // console.log(data_nasc);
```

```

    localStorage.setItem('userCadSexo', result[0].sexo);
    localStorage.setItem('userCadData', data_nasc);
}

```

backend/user\_sus/relatorio\_usuario.js

```

const btniniciar_atend = document.querySelector("#iniciar-atend");
const btnrelatorio_consult = document.querySelector("#relatorio-ulti");
function iniciar_atendimento() {
    window.location = "../pages/menu2.html";
}
function Ultima_consulta() {
    window.location = "../pages/relatorioConsulta.html";
}
btniniciar_atend.addEventListener("click", iniciar_atendimento);
btnrelatorio_consult.addEventListener("click", Ultima_consulta);

```

backend/user\_sus/seleciona.js

```

function seleccionarUserSus(userSus) {
    let idUserSus = userSus.value;
    // console.log(cartSus);
    localStorage.setItem('idUserSus', idUserSus);
    //window.location = "../pages/menu2.html";
    window.location = "../pages/Menu_intermediario.html";
}

```

css/cadastro.css

```

*{
    box-sizing: border-box;
    padding: 0;
    margin: 0;
}
.fundo-cadastro-global{
    height: 100vh;
    display: flex;
    flex-direction: column;
    align-items: center;
}
.fundo-cadastro{
    width: 70%;
    height: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
.fundo-cadastro .img img{
    padding: 5px;
    height: 150px;
    margin: 20px;
}

```

```

}
.fundo-cadastro form{
  width: 100%;
  padding: 5px;
  margin: 10px 0px;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}
.fundo-cadastro form div{
  width: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.fundo-cadastro .form{
  width: 100%;
  text-align: center;
}
.fundo-cadastro .btn-cad-novo-user{
  text-align: center;
  width: 50%;
}
.fundo-cadastro input{
  border: none;
  border-bottom: 2px solid rgb(177, 177, 177);
  padding: 10px 5px;
  margin: 10px 0px;
  width: 90%;
}
.fundo-cadastro input:focus{
  border-bottom: 2px solid rgb(98, 98, 98);
  outline: none;
}
.fundo-cadastro .input-radio{
  display: flex;
  flex-direction: row;
  align-items: flex-start;
}
.fundo-cadastro .input-radio div{
  display: flex;
  flex-direction: row;
  padding: 20px 0px;
}
.fundo-cadastro .input-radio input{
  width: 20px;
}

```

```

}
.fundo-cadastro .input-radio label{
  font-weight: 500;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
}
.fundo-cadastro button{
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  margin: 5px 0px;
  width: 90%;
  border: none;
  background-color: #80cc80;
  box-shadow: 1px 1px 2px rgb(83, 83, 83);
  padding: 1.1rem 0px;
  font-size: 1rem;
  font-weight: 700;
}
.fundo-cadastro button:hover{
  background-color: #80cc80d4;
  box-shadow: 1px 2px 3px rgb(48, 48, 48);
}
.fundo-cadastro .form #enabled-case-admin{
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  background-color: transparent;
}
.fundo-cadastro .form #enabled-case-admin button{
  font-size: .9rem !important;
  width: 45% !important;
}
.btn-ajuda{
  width: 90%;
  display: flex;
  justify-content: flex-end;
}
.btn-ajuda #btn-ajuda{
  background-color: transparent;
  width: 35%;
  padding: .3rem;
  border-radius: 10px;
  border: 1px solid rgb(190, 190, 190);
  box-shadow: 2px 2px 2px rgb(160, 160, 160);
  font-size: .9rem;
  display: flex;
  flex-direction: column;
  align-items: center;
}
.btn-ajuda #btn-ajuda:hover{

```

```

    background-color: #80cc80d4;
    box-shadow: 4px 4px 4px rgb(160, 160, 160);

}
#retorno-ajuda-user{
    padding: 1rem 0rem;
}

/*Formatação dos dados de retorno*/
.fundo-cadastro #result{
    display: flex;
    flex-direction: column;
    align-items: center;
    width: 80%;

}
.fundo-cadastro #result .equipe{
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-size: .9rem;
    color: black;
    margin: 1px 0 !important;
    background-color: rgb(201, 201, 201);

}
.fundo-cadastro #result .equipe:hover{
    background-color: rgb(168, 168, 168);

}

/*Formatação para o footer*/
.fundo-cadastro-global .footer{
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: flex-end;

}
.fundo-cadastro-global .footer .footer-logos{
    display: flex;
    justify-content: flex-end;

}
.fundo-cadastro-global .footer .footer-logos :nth-child(1){
    width: 40%;

}
.fundo-cadastro-global .footer h4{
    color: rgb(70, 70, 70);
    font-size: 1rem;
    padding: 5px 10px;

}
.fundo-cadastro-global .footer a{
    text-decoration: none;
    padding: 5px;
    color: rgb(0, 0, 0);

```

```

}
.fundo-cadastro-global .footer a:hover{
    color: #80cc80;

}
.fundo-cadastro-global .footer img{
    width: 70px;
    height: 70px;
    margin: 5px;

}
/*Formatação para o menu global*/
/*
.menu{
    width: 100%;
    background-color: #a5d6a7;
    box-shadow: 0px 5px 5px rgb(133, 133, 133);

}
.menu .navbar-collapse div{
    width: 100%;
    display: flex;
    justify-content: space-between;
    padding: 0 50px;

}
.menu .navbar-nav{
    display: flex;
    justify-content: space-around;
    align-items: center;
    width: 100%;

}
.menu .nav-item{
    padding: 0px 5px;
    margin: 1px 5px;

}
.menu .navbar-nav a{
    color: black;
    font-weight: 500;

}
*/

```

css/displayNone.css

```

/*Disable textos*/
.text-none-aspectos,
.text-none-proped,
.text-none-sinais,
.text-none-rastreamento,
.text-none-antropom,
.text-none-estrat-risco{
    display: none;

}

```



```

/*Disable textos internos*/
.text-none-imc,
.text-none-circ-abdom,
.text-none-cint-quadr,
.text-none-torn-braq{
    display: none;

}
.text-none-obj-entrevista,
.text-none-quest-import{
    display: none;

}

/*Disable etapa 2 'estratificação de risco'*/
.disable-etapa-2,
.disable-etapa-3{
    display: none;

}

/*Disable caso login não 'admin'*/
.enabled-case-admin{
    display: none;

}

/*
.disable-btn-protocol,
.disable-btn-busca,
.disable-btn-consulta{
    display: none;

}
*/
/*
.disable-btn-aval{
    display: none !important;

}
*/

```

css/global.css

```

/*Formatação global*/
*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;

}
body{
    height: 100vh;
    margin: 0;
    background-color: rgb(236, 236, 236);
}

```

```

}
.fundo1{
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  height: 100%;
}

/*Formatação para o menu global*/
.menu{
  background-color: #a5d6a7;
  box-shadow: 0px 2px 2px rgb(180, 180, 180);
  border-bottom: 1px solid rgb(138, 138, 138);
  width: 100%;
}
.menu .navbar-collapse div{
  width: 100%;
}
.menu .navbar-nav{
  display: flex;
  justify-content: space-between;
  width: 100%;
  padding: 0px 10px;
}
.menu .nav-item{
  padding: 5px;
  margin: 0px 5px;
}
.menu .nav-item .nav-link{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.menu .nav-item::marker{
  font-size: 0rem;
}
.menu .navbar-nav a{
  color: black;
  font-weight: 500;
}
.menu .icons-menu{
  width: 3rem;
  border-radius: 10px;
  padding: 5px;
}
.menu .text-menu{
  text-align: left;
}

```

```

    font-size: 10px;

}
.menu .listra-menu{
    text-align: middle;
    font-size: 10px;
    border-bottom: 2px solid red;
}

.menu .icons-menu:hover{
    background-color: #80cc80c4;
}

/*Formatção sub menu*/
.sub-menu-necess{
    background-color: #a5d6a7;
    box-shadow: 0px 5px 5px rgb(180, 180, 180);
    border-top: 1px solid rgb(78, 78, 78);
}

.sub-menu-necess .navbar-collapse div{
    width: 100%;
}

.sub-menu-necess .navbar-nav{
    width: 100%;
    justify-content: space-evenly;
}

.sub-menu-necess .icons-menu{
    width: 3.1rem;
    padding: 5px;
    border-radius: 10px;
}

.sub-menu-necess .icons-menu:hover{
    background-color: #80cc80d4;
}

.sub-menu-necess .nav-item .nav-link{
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

/*Formatação padrão para os botões globais*/
.fundo-botao{
    padding: 1.5rem 0 1rem 0;
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: center;
    width: 100%;
}

```

```

.botao-padrao{
  text-decoration: none;
  margin: 10px 2px;
  width: 90%;
  text-align: center;
}

.botao-padrao button{
  border: none;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  padding: 2rem .2rem;
  font-size: 1.5rem;
  font-weight: 600;
  color: black;
  background-color: #80cc80;
  box-shadow: 0px 0px 7px rgba(0, 0, 0, 0.6);
  border: 1px solid rgb(150, 150, 150);
  width: 100%;
  cursor: pointer;
}

.botao-padrao button:hover{
  background-color: #80cc80d4;
}

.icon-expand{
  padding: .2rem .5rem;
}

.flag-risco img{
  width: 3%;
  margin: 2px 5px;
}

.active{
  background-color: #43a443;
}

/*Formatação para os textos escondidos*/
.text-none-style, .text-none-tec-proped{
  background-color: white;
  padding: 1.5rem 1rem;
  margin: 5px 0px;
  width: 85%;
  color: black;
  font-weight: 500;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  text-align: center;
  font-size: 1.1rem;
  box-shadow: 1px 1px 10px rgb(72, 72, 72);
}

.text-none-style p,
.text-none-tec-proped p{
  padding: 2px 0px;
}

```

```

}
.text-none-style h4{
    padding: 10px 0px;
    font-weight: 600;
    font-size: 1.1rem;

}
.text-none-style .ajuste-text p,
.text-none-style .ajuste-text h5{
    text-align: left;

}
.text-none-style .ajuste-text
.ajuste-text-destaque{
    padding: 5px;
    border: 1px solid black;
    box-shadow: 0px 0px 7px rgb(102, 102, 102);

}
/*Formatação para inputs globais*/
.form-style{
    display: flex;
    flex-direction: column;
    width: 100%;

}
.form-style input{
    width: 60%;
    height: 50px;
    padding: 5px;
    margin: 5px;
    border: 1px solid rgb(150, 150, 150);
    box-shadow: 0px 0px 5px rgb(150, 150, 150);
    background-color: rgba(219, 219, 219, 0.5);

}
.form-style input::placeholder{
    font-weight: bold;
    color: rgba(0, 0, 0, 0.5);

}
.form-style button{
    border: none;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    padding: 15px 0px;
    margin: 20px 0;
    font-size: 1rem;
    font-weight: 600;
    color: black;
    width: 50%;
    border: 1px solid rgb(150, 150, 150);
    background-color: #80cc80;
    box-shadow: 0px 0px 5px rgb(150, 150, 150);

}
.form-style button:hover{

```

```

        background-color: #80cc80d4;

    }

    /*Formatação de botões internos*/
    .text-none-style .flex-ajuste{
        display: flex;
        flex-direction: column;
        align-items: center;
        width: 100%;

    }
    .text-none-style-interno{
        padding: 5px 5px;
        margin: 5px 5px;
        width: 85%;
        color: black;
        font-weight: 500;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        text-align: center;
        font-size: 1rem;
        box-shadow: 1px 1px 10px rgb(72, 72, 72);

    }
    .text-none-style-interno div{
        display: flex;
        flex-direction: column;
        align-items: center;

    }
    .botao-padrao-interno{
        margin: 7px 0px;
        width: 90%;

    }
    .botao-padrao-interno button{
        width: 100%;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        font-size: 1rem;
        font-weight: 600;
        background-color: #80cc80;
        box-shadow: 0px 0px 7px rgba(0, 0, 0, 0.6);
        border: 1px solid rgb(150, 150, 150);

    }
    .form-style-interno{
        display: flex;
        flex-direction: column;
        width: 100%;

    }
    .form-style-interno div{
        display: block;

    }
    .form-style-interno input{
        width: 60%;

```

```

height: 50px;
border: none;
padding: 5px;
margin: 5px;
border: 1px solid rgb(150, 150, 150);
box-shadow: 0px 0px 5px rgb(150, 150, 150);
background-color: rgba(219, 219, 219, 0.5);
}
.form-style-interno input::placeholder{
  font-weight: bold;
  color: rgba(0, 0, 0, 0.5);
}
.form-style-interno button{
  border: none;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  padding: 15px 0px;
  margin: 20px 0;
  font-size: 1rem;
  font-weight: 600;
  color: black;
  width: 50%;
  border: 1px solid rgb(150, 150, 150);
  background-color: #80cc80;
  box-shadow: 0px 0px 5px rgb(150, 150, 150);
}
.form-style-interno button:hover{
  background-color: #80cc80d4;
}

/*Formatação especifica para inputs checkbox*/
.form-style-checkbox{
  display: flex;
  flex-direction: column;
  justify-content: start;
  align-items: flex-start;
}
.form-style-checkbox h5{
  font-weight: 600;
  padding: 20px 0px 5px 0px;
  font-size: 1rem;
}
.form-style-checkbox h6{
  font-weight: 600;
  font-size: 1rem;
}
.form-style-checkbox .form-style{
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  padding: 10px 0px;
}

```

```

}
.form-style-checkbox .style-checkbox{
  display: flex;
  align-items: center;
  margin: 7px 0px;
}
.style-checkbox input{
  width: 20px;
  height: 20px;
  margin: 5px;
  box-shadow: none;
}
.style-checkbox label{
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-weight: 600;
  font-size: .8rem;
  text-align: left;
}

/*Formatação das paginas só com texto*/
.text-style-pure{
  padding: 5rem 3rem;
  height: 100%;
}
.text-style-pure p{
  text-align: justify;
  font-size: 1.2rem;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-weight: 500;
}
.text-style-pure2{
  padding: 1rem 1rem 2.5rem 1rem;
}
.text-style-pure2 p{
  text-align: justify;
  font-size: 1rem;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-weight: 500;
  padding: 0;
}
.text-style-pure2 .sub-title{
  font-weight: 600;
  text-align: justify;
}

/*Formatação estratificação etapa 2*/
.etapa-2{
  display: flex;

```



```

        justify-content: space-between;
        width: 100%;
        margin: 20px 0px;
        padding: 20px 0px;
        border-top: 1px solid black;
        border-bottom: 1px solid black;
    }

    .etapa-2 h5{
        font-weight: 600;
        font-size: 1rem;
    }

    }

    .etapa-2 .etapa-2-form{
        width: 40%;
        display: flex;
        flex-direction: column;
        align-items: flex-start;
        justify-content: center;
    }

    }

    .etapa-2 .etapa-2-form form{
        display: flex;
        flex-direction: column;
    }

    }

    .etapa-2 .etapa-2-form form div{
        width: 100%;
        display: flex;
        align-items: flex-start;
    }

    }

    .etapa-2 .etapa-2-form input{
        width: 100%;
    }

    }

    .etapa-2 .etapa-2-form input::placeholder{
        font-weight: bold;
        color: rgba(0, 0, 0, 0.5);
    }

    }

    .etapa-2 .etapa-2-form .style-checkbox{
        width: 30%;
        height: 30%;
        display: flex;
        align-items: center;
    }

    }

    .etapa-2 .etapa-2-form button{
        width: 100%;
    }

    }

    .etapa-2 .etapa-2-img{
        width: 60%;
    }

    }

    /*Formatação estratificação etapa 3*/

```

```

.etapa-3{
  width: 100%;
  margin: 50px 0px;
}
.etapa-3 h5{
  font-weight: 600;
  font-size: 1rem;
}

/*Formatação pagina hemograma*/
.form-style-hemograma{
  display: flex;
  align-items: center;
  justify-content: center;
  width: 70%;
}
.form-style-hemograma h5{
  font-weight: 700;
  font-size: 1.5rem;
  padding: 0px 5px 20px 5px;
}
.form-style-hemograma p{
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-size: 1rem;
  color: black;
  font-weight: bold;
}
.form-style-hemograma div{
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 100%;
  padding: 15px 5px;
}
.form-style-hemograma label{
  font-weight: 600;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
}
.form-style-hemograma input{
  width: 100%;
}
.form-style-hemograma button{
  width: 80%;
}

/*Formatação botão finalizar*/
.fundo-botao .botao-padrao #btn-finaliza-consulta,
.fundo-botao .botao-padrao #btn-finaliza-ava-clinic{

```

```

font-family: Verdana, Geneva, Tahoma, sans-serif;
padding: 1.3rem 1rem;
margin: 10px 0;
font-size: 1rem;
}
.fundo-botao .botao-padrao #btn-finaliza-consulta:hover,
.fundo-botao .botao-padrao #btn-finaliza-ava-clinic:hover{
    background-color: #558a55;
}

/*Formatação navegação rodapé*/
.navegacao-footer{
    display: flex;
    justify-content: space-around;
}
.navegacao-footer .botao-padrao{
    width: 35%;
    margin: 5px 0 10px 0;
}
.navegacao-footer .botao-padrao button{
    background-color: transparent;
    box-shadow: 2px 2px 2px rgb(160, 160, 160);
    border: 2px solid rgb(122, 122, 122);
    padding: 1rem 0;
    font-size: 1.3rem;
}
.navegacao-footer .botao-padrao button:hover{
    background-color: #80cc80d4;
    box-shadow: 4px 4px 4px rgb(160, 160, 160);
}

/*Formatação pagina relatorio*/
.fundo-botao-print{
    display: flex;
    height: 100%;
    flex-direction: row;
    align-items: flex-start;
    justify-content: center;
}
.fundo-botao-print div{
    width: 50%;
    display: flex;
    flex-direction: column;
    align-items: center;
}
.fundo-botao-print .col-consultas{
    border-right: 2px solid rgb(160, 160, 160);
}
.fundo-botao-print .col-consultas{

```

```

border-left: 2px solid rgb(160, 160, 160);

}
.fundo-botao-print .col-consultas div{
width: 95%;
display: flex;
flex-direction: column;
align-items: center;
justify-content: flex-start;
padding: 1rem 0;

}
.fundo-botao-print .col-consultas form{
width: 100%;
display: flex;
flex-direction: column;
align-items: center;
padding: 0;

}
.fundo-botao-print .col-consultas form input{
border: none;
width: 100%;
height: 4rem;
padding: 5px;
border: 1px solid rgb(195, 195, 195);
border-radius: 5px;

}
.fundo-botao-print .col-consultas form input::placeholder{
color: rgb(118, 118, 118);
font-family: Verdana, Geneva, Tahoma, sans-serif;

}
.fundo-botao-print .col-relatorios div{
width: 95%;
display: flex;
flex-direction: column;
align-items: center;
justify-content: flex-start;
padding: 1rem 0;

}
.fundo-botao-print button{
width: 100%;
font-size: 1.1rem;
padding: 1.3rem 0;
font-family: Verdana, Geneva, Tahoma, sans-serif;

}
.fundo-botao-print .form-select{
height: 4rem;

}
.fundo-botao-print .form-select option{
color: black;
background-color: rgb(230, 230, 230);

```

```

font-family: Verdana, Geneva, Tahoma, sans-serif;
font-size: .9rem;
font-weight: 500;
}
.fundo-botao-print .retorno-select-1 table{
font-family: Verdana, Geneva, Tahoma, sans-serif;
font-size: .8rem;
width: 100%;
border-radius: 10px;
box-shadow: 3px 3px 4px rgb(113, 113, 113);
}
.retorno-select-1 th{
background-color: rgb(220, 220, 220);
color: black;
padding: 10px 0;
font-weight: 600;
border: none;
text-align: center;
}
.retorno-select-1 td{
background-color: rgb(229, 229, 229);
color: black;
padding: 10px 0;
font-weight: 500;
border: none;
text-align: center;
}
}

/*Formatação pagina direitos*/
.pageDireitos{
display: flex;
flex-direction: column;
justify-content: flex-start;
}

.pageDireitos h1 {
font-weight: bold;
text-align: center;
width: 30%;
}

.pageDireitos h4 {
font-weight: bold;
text-align: left;
width: 60%;
}

.pageDireitos h5 {
text-align: justify;
width: 60%;
}

/*fomatação botao pagina relatorios*/

```

```
.botao-relatorio{
  padding-top: 20px;
}
```

css/index.css

```
/* CSS reset */
*, *:before, *:after {
  margin:0;
  padding:0;
  font-family: Arial,sans-serif;
}

body{
  margin:10px;
}
#idlogo {
  height: 100px;
}

a{
  text-decoration: none;
}

/* esconde as ancoras da tela */
a.links{
  display: none;
}

/* content que contem os formulários */
.content{
  width: 600px;
  margin: 0px auto;
  position: relative;
}

/* formatando o cabeçalho dos formulários */
h1{
  font-size: 48px;
  color: #066a75;
  padding: 10px 0;
  font-family: Arial,sans-serif;
  font-weight: bold;
  text-align: center;
  padding-bottom: 30px;
}
h1:after{
  content: ' ';
  display: block;
  width: 100%;
  height: 2px;
  margin-top: 10px;
  background: -webkit-linear-gradient(left, rgba(147,184,189,0) 0%,rgba(147,184,189,0.8)
20%,rgba(147,184,189,1) 53%,rgba(147,184,189,0.8) 79%,rgba(147,184,189,0) 100%);
  background: linear-gradient(left, rgba(147,184,189,0) 0%,rgba(147,184,189,0.8)
20%,rgba(147,184,189,1) 53%,rgba(147,184,189,0.8) 79%,rgba(147,184,189,0) 100%);
}
```

```

}

p{
  margin-bottom:15px;
}
p:first-child{
  margin: 0px;
}
label{
  color: #405c60;
  position: relative;
}

/**** advanced input styling ****/
/* placeholder */
::-webkit-input-placeholder {
  color: #bebcbc;
  font-style: italic;
}
input:-moz-placeholder,
textarea:-moz-placeholder{
  color: #bebcbc;
  font-style: italic;
}
input {
  outline: none;
}

input:not([type="checkbox"]){
  width: 95%;
  margin-top: 4px;
  padding: 10px;
  border: 1px solid #b2b2b2;

  -webkit-border-radius: 3px;
  border-radius: 3px;

  -webkit-box-shadow: 0px 1px 4px 0px rgba(168, 168, 168, 0.6) inset;
  box-shadow: 0px 1px 4px 0px rgba(168, 168, 168, 0.6) inset;

  -webkit-transition: all 0.2s linear;
  transition: all 0.2s linear;
}

/*estilo do botão submit */
input[type="submit"]{
  width: 100%!important;
  cursor: pointer;
  background: #3d9db3;
  padding: 8px 5px;
  color: #fff;
  font-size: 20px;
  border: 1px solid #fff;
  margin-bottom: 10px;
  text-shadow: 0 1px 1px #333;

  -webkit-border-radius: 5px;

```

```

border-radius: 5px;

transition: all 0.2s linear;
}
input[type="submit"]:hover{
    background: #4ab3c6;
}

/*marcando os links para mudar de um formulário para outro */
.link{
    position: absolute;
    background: #e1eaeb;
    color: #7f7c7c;
    left: 0px;
    height: 20px;
    width: 440px;
    padding: 17px 30px 20px 30px;
    font-size: 16px;
    text-align: right;
    border-top: 1px solid #dbe5e8;

    -webkit-border-radius: 0 0 5px 5px;
    border-radius: 0 0 5px 5px;
}
.link a {
    font-weight: bold;
    background: #f7f8f1;
    padding: 6px;
    color: rgb(29, 162, 193);
    margin-left: 10px;
    border: 1px solid #cbd518;

    -webkit-border-radius: 4px;
    border-radius: 4px;

    -webkit-transition: all 0.4s linear;
    transition: all 0.4s linear;
}
.link a:hover {
    color: #39bfd7;
    background: #f7f7f7;
    border: 1px solid #4ab3c6;
}

/* estilos para para ambos os formulários */
#cadastro,
#login,
#menu{
    position: absolute;
    top: 0px;
    width: 88%;
    padding: 18px 6% 60px 6%;
    margin: 0 0 35px 0;
    background: rgb(247, 247, 247);
    border: 1px solid rgba(147, 184, 189,0.8);

    -webkit-box-shadow: 5px;

```



```

border-radius: 5px;

-webkit-animation-duration: 0.5s;
-webkit-animation-timing-function: ease;
-webkit-animation-fill-mode: both;

animation-duration: 0.5s;
animation-timing-function: ease;
animation-fill-mode: both;
}

#paracadastro:target ~ .content #cadastro,
#paralogin:target ~ .content #login{
  z-index: 2;
  -webkit-animation-name: fadeInLeft;
  animation-name: fadeInLeft;

  -webkit-animation-delay: .1s;
  animation-delay: .1s;
}
#registro:target ~ .content #login,
#paralogin:target ~ .content #cadastro{
  -webkit-animation-name: fadeOutLeft;
  animation-name: fadeOutLeft;
}

/*fadeInLeft*/
@-webkit-keyframes fadeInLeft {
  0% {
    opacity: 0;
    -webkit-transform: translateX(-20px);
  }
  100% {
    opacity: 1;
    -webkit-transform: translateX(0);
  }
}

@keyframes fadeInLeft {
  0% {
    opacity: 0;
    transform: translateX(-20px);
  }
  100% {
    opacity: 1;
    transform: translateX(0);
  }
}

/*fadeOutLeft*/
@-webkit-keyframes fadeOutLeft {
  0% {
    opacity: 1;
    -webkit-transform: translateX(0);
  }
  100% {
    opacity: 0;

```

```
-webkit-transform: translateX(-20px);
}
}
```

```
@keyframes fadeOutLeft {
  0% {
    opacity: 1;
    transform: translateX(0);
  }
  100% {
    opacity: 0;
    transform: translateX(-20px);
  }
}
```

css/necessidades.css

```
.form-necessidades{
  width: 80%;
  padding: 10px;
}
.form-necessidades p{
  font-weight: 600;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-size: 1rem;
  padding: 20px 0px 0px 0px;
}
.form-necessidades .style-checkbox{
  display: flex;
  flex-direction: row;
  align-items: center;
  padding: 2px 0px;
}
.form-necessidades button{
  width: 100%;
  margin-top: 30px;
}
}
```

css/relatorio.css

```
.body-relatorio{
  border: solid;
  padding: 20px;
  width: 80%;
  margin: 20px auto 0px auto;
}

.conteudo_relatorio {
  margin-top: 20px;
  margin-bottom: 20px;
}

.titulo-pdf{
```

```

    margin-bottom: 10px;
    text-align: center;
    font-style: bold;
}

.retorno-select-2 {
    display: flex;
    justify-content: center;
}

.retorno-select-2 table{
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-size: 1rem;
    border-radius: 10px;
    box-shadow: 3px 3px 4px rgb(113, 113, 113);
    width:80%;
}

.retorno-select-2 th,
.retorno-select-2 td{
    background-color: rgb(169, 169, 169);
    color: black;
    padding: 10px 10px;
    border: none;
    text-align: center;
}

.retorno-select-2 td{
    background-color: rgb(229, 229, 229);
}

/* Específico da página do relatório da última consulta */
.retorno-select-3 {
    text-align: left;
}

.retorno-select-3 h3{

    text-align: center;
}

.info {
    border-top: 2px solid black;
    margin: 20px 100px;
    padding: 12px 0;
    font-size: 1.2em;
}

.infos {
    display: flex;
    justify-content: space-evenly;
    text-align: justify;
    margin-top: 12px;
}

.exames{
    margin-right: 12px;
    text-align: left;
}

```

db/db\_assist\_enferm\_4.sql

-- MySQL dump 10.13 Distrib 8.0.31, for Win64 (x86\_64)

--

-- Host: localhost Database: assist\_enferm

-----

-- Server version 8.0.31

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

--

-- Table structure for table `consulta`

--

```
DROP TABLE IF EXISTS `consulta`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `consulta` (
  `data_cons` datetime NOT NULL,
  `fk_id_usersus` int NOT NULL,
  `pas` float DEFAULT NULL,
  `clas_ha` varchar(15) DEFAULT NULL,
  `imc` float DEFAULT NULL,
  `clas_imc` varchar(20) DEFAULT NULL,
  `clas_abdo` varchar(20) DEFAULT NULL,
  `clas_cint_quad` varchar(15) DEFAULT NULL,
  `clas_torn_brac` varchar(60) DEFAULT NULL,
  `indicadores` text,
  `clas_estrat` varchar(15) DEFAULT NULL,
  `list_psicobio` text,
  `list_psicosoc` text,
  `list_psicoesp` text,
  PRIMARY KEY (`data_cons`,`fk_id_usersus`),
  KEY `fk_cons_usersus_idx` (`fk_id_usersus`),
  CONSTRAINT `fk_cons_usersus` FOREIGN KEY (`fk_id_usersus`) REFERENCES `user_sus` (`id_usersus`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

--

-- Dumping data for table `consulta`

--

```
LOCK TABLES `consulta` WRITE;
/*!40000 ALTER TABLE `consulta` DISABLE KEYS */;
INSERT INTO `consulta` VALUES ('2022-12-04 16:17:31',1,100,'HA estágio
2',NULL,NULL,NULL,NULL,NULL,'ALTO RISCO',NULL,NULL,NULL),('2022-12-04
16:54:00',1,100,NULL,NULL,NULL,NULL,NULL,NULL,'BAIXO RISCO',NULL,NULL,NULL),('2022-12-
```

```

04 16:57:00',1,100,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'ALTO RISCO',NULL,NULL,NULL),('2022-
12-11 21:11:00',2,100,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'ALTO
RISCO',NULL,NULL,NULL),('2022-12-12
20:17:48',1,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,'NECESSIDADES
PSICOBIOLÓGICAS:\n- Oxigenação:\nPadrão respiratório prejudicado.\nEnsinar técnica de respiração
lenta (10 respirações a cada minuto, durante 15 minutos,\n          uma vez ao dia).\nVerificar e
registrar sinais vitais.\n',NULL,NULL);
/*!40000 ALTER TABLE `consulta` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `hemograma`
--

DROP TABLE IF EXISTS `hemograma`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `hemograma` (
  `id_hemo` int NOT NULL AUTO_INCREMENT,
  `fk_id_usersus` int NOT NULL,
  `fk_data_cons` datetime NOT NULL,
  `hemacias` float DEFAULT NULL,
  `hemoglobina` float DEFAULT NULL,
  `hematocrito` float DEFAULT NULL,
  `leucocitos` float DEFAULT NULL,
  `linfocitos` float DEFAULT NULL,
  `plaquetas` float DEFAULT NULL,
  `glicemia_jejum` float DEFAULT NULL,
  `colesterol_total` float DEFAULT NULL,
  `hdl` float DEFAULT NULL,
  `triglicerideos` float DEFAULT NULL,
  `hemoglobina_glicada` float DEFAULT NULL,
  `creatina` float DEFAULT NULL,
  `potassio_plasmatico` float DEFAULT NULL,
  `equ` float DEFAULT NULL,
  `microalbumina_urina` float DEFAULT NULL,
  `albumina` float DEFAULT NULL,
  `acido_urico` float DEFAULT NULL,
  `ureia` float DEFAULT NULL,
  `tgo` float DEFAULT NULL,
  `tgp` float DEFAULT NULL,
  PRIMARY KEY (`id_hemo`,`fk_id_usersus`,`fk_data_cons`),
  KEY `fk_id_usersus` (`fk_id_usersus`,`fk_data_cons`),
  CONSTRAINT `hemograma_ibfk_1` FOREIGN KEY (`fk_id_usersus`,`fk_data_cons`) REFERENCES
`consulta` (`fk_id_usersus`,`data_cons`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `hemograma`
--

LOCK TABLES `hemograma` WRITE;
/*!40000 ALTER TABLE `hemograma` DISABLE KEYS */;
INSERT INTO `hemograma` VALUES (1,1,'2022-12-04
16:17:31',4.58,14.2,41.8,8300,39,327000,89,206,84,75,5.6,0.8,4.2,13,18,4.8,2.8,34,20,40),(3,1,'2022-
12-12

```

```
20:17:48',NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NUL
L,NULL,NULL,NULL,NULL);
/*!40000 ALTER TABLE `hemograma` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `user_sus`
--
```

```
DROP TABLE IF EXISTS `user_sus`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user_sus` (
  `id_usersus` int NOT NULL AUTO_INCREMENT,
  `cart_sus` varchar(15) NOT NULL,
  `cpf` varchar(11) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `data_nasc` date NOT NULL,
  `sexo` varchar(10) NOT NULL,
  `fk_id_equipe` int NOT NULL,
  PRIMARY KEY (`id_usersus`),
  KEY `fk_id_equipe` (`fk_id_equipe`),
  CONSTRAINT `user_sus_ibfk_1` FOREIGN KEY (`fk_id_equipe`) REFERENCES `usuario` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `user_sus`
--
```

```
LOCK TABLES `user_sus` WRITE;
/*!40000 ALTER TABLE `user_sus` DISABLE KEYS */;
INSERT INTO `user_sus` VALUES (1,'123456789012345','12345678901','Lucas','2001-06-
06','Masculino',2),(2,'11122233334444','11122233344','Isabela','2002-07-
02','Feminino',2),(3,'000111122223333','00011122233','Rodrigo','1990-06-06','Masculino',2);
/*!40000 ALTER TABLE `user_sus` ENABLE KEYS */;
UNLOCK TABLES;
```

```
--
-- Table structure for table `usuario`
--
```

```
DROP TABLE IF EXISTS `usuario`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `usuario` (
  `id` int NOT NULL AUTO_INCREMENT,
  `login` varchar(100) NOT NULL,
  `senha` varchar(100) NOT NULL,
  `membros` tinytext,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
-- Dumping data for table `usuario`
--
```

```

LOCK TABLES `usuario` WRITE;
/*!40000 ALTER TABLE `usuario` DISABLE KEYS */;
INSERT INTO `usuario` VALUES
(1,'admin','admin123','SuperUser'),(2,'equipe1','12345','LucasWinglissonEduardo'),(3,'equipe3','11111',
'Mauro da Nobrega\nJuliana Lima de Souza\nPaula Zacian\nRoger
Machado'),(4,'equipe4','123','Caralá'),(5,'equipe5','123','Lucas'),(6,'equipe6','444','Winglisson');
/*!40000 ALTER TABLE `usuario` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2022-12-12 20:32:44

```

js/Utils/calculos.js

//CALCULOS DO PROJETO;

```

const btnPasPad = document.querySelector("#btn-calcular-pas-pad");
const btnCirAbd = document.querySelector("#btn-circ-abdom-calcular");
const btnCintQuad = document.querySelector("#btn-cint-quadr-calcular");
const btnTornBraq = document.querySelector("#btn-torn-braq-calcular");
const btnEstrat2Calcular = document.querySelector("#btn-estrat2-calcular");

```

```

function calcular_pas_pad() {
    let pas = document.querySelector("#pas").value;
    let pad = document.querySelector("#pad").value;
    let result = document.querySelector("#retorno-calculo-pas-pad");

    let ha = 'Valor indeterminado';

    if (pas >= 180 || pad >= 110) {
        ha = 'HA estágio 3';
    }
    else if (pas >= 160 || pad >= 100) {
        ha = 'HA estágio 2';
    }
    else if (pas >= 140 || pad >= 90) {
        ha = 'HA estágio 1';
    }
    else if (pas >= 130 || pad >= 85) {
        ha = 'Pré-hipertensão';
    }
    else if (pas >= 120 || pad >= 80) {
        ha = 'PA normal';
    }
    else { //pas < 120 AND pad < 80
        ha = 'PA ótima'
    }
}

```

```

        localStorage.setItem('clas_ha', ha);
        result.innerHTML = ha;
    }

    /*
    function calculo_imc()
    {
        let peso = document.querySelector("#imc-peso").value;
        let altura = document.querySelector("#imc-altura").value;
        let imc = peso/(altura**2);
        let resultado_imc = document.querySelector("#retorno-calculo-imc");
        resultado_imc.innerHTML = imc;
    }*/

    function classif_circ_abdominal() {

        let sexo = localStorage.getItem('userCadSexo');
        let circ_abdom = document.querySelector("#circ-abdom-cm").value;
        let classificacao;
        let resultado_circ = document.querySelector("#retorno-calculo-circ-abdom");
        if (sexo == 'Masculino') {
            if (circ_abdom < 94) {
                classificacao = 'Normal';
            }
            else if (circ_abdom < 102) {
                classificacao = 'Risco Elevado';
            }
            else {
                classificacao = 'Risco Muito Elevado';
            }
        }
        else {
            if (circ_abdom < 80) {
                classificacao = 'Normal';
            }
            else if (circ_abdom < 88) {
                classificacao = 'Risco Elevado';
            }
            else {
                classificacao = 'Risco Muito Elevado';
            }
        }
        localStorage.setItem('clas_abdo', classificacao);
        resultado_circ.innerHTML = classificacao;
    }

    function cint_quadril() {
        let circun_abdmon = document.querySelector("#cint-quadr-abdom-cm").value;
        let circun_quadril = document.querySelector("#cint-quadr-quadr-cm").value;
        let result;
        let result_final = document.querySelector("#retorno-calculo-cint-quadr");
        let sexo = localStorage.getItem('userCadSexo');

        result = circun_abdmon / circun_quadril;
        localStorage.setItem('cint_quad', result);
    }

```



```

    if (sexo == 'Masculino') {
        if (result > 0.9) {
            result = 'Risco elevado';
        }
        else {
            result = 'Normal';
        }
    }
    else {
        if (result > 0.85) {
            result = 'Risco elevado';
        }
        else {
            result = 'Normal';
        }
    }
    localStorage.setItem('clas_cint_quad', result);
    result_final.innerHTML = result;
}

function torno_braq() {
    let tornozelo = parseInt(document.querySelector("#torn-braq-pas-torn").value);
    let braquial = parseInt(document.querySelector("#torn-braq-pas-bra").value);
    let result_torno_braquial;
    let result_final_tornozelobraquial = document.querySelector("#retorno-calculo-torn-braq");

    result_torno_braquial = tornozelo / braquial;
    localStorage.setItem('torn_braq', result_torno_braquial);
    if (result_torno_braquial < 0.9) {
        result_torno_braquial = 'Presença de Doença Obstrutiva';
    }
    else if (result_torno_braquial <= 0.99) {
        result_torno_braquial = 'Limítrofe';
    }
    else if (result_torno_braquial <= 1.4) {
        result_torno_braquial = 'Normal';
    }
    else {
        result_torno_braquial = 'Incompressibilidade arterial devido a provável calcificação';
    }
    localStorage.setItem('clas_torn_braq', result_torno_braquial);
    result_final_tornozelobraquial.innerHTML = result_torno_braquial;
}

function calcularEstratificacaoParte2() {

    let pontos = 0;
    let idade = localStorage.getItem('userCadIdade')
    let colesterol = document.querySelector('#etapa2-colest-total').value;
    let fumante = document.querySelector('#etapa2-fumante')
    let pas2 = document.querySelector('#etapa2-pa-sistol').value;
    let tratada = document.querySelector('#etapa2-tratada')
    let retorno_calculo_estratificacao = document.querySelector('#retorno-calculo-estrat3')
    let perc_risco = "..."
    if ((localStorage.getItem('userCadSexo')) === 'Masculino') {

```

```

if ((idade > 20) && (idade <= 34)) { pontos += -9 }
else if ((idade > 34) && (idade <= 39)) { pontos += -4 }
else if ((idade > 40) && (idade <= 44)) { pontos += 0 }
else if ((idade > 45) && (idade <= 49)) { pontos += 3 }
else if ((idade > 50) && (idade <= 54)) { pontos += 6 }
else if ((idade > 55) && (idade <= 59)) { pontos += 8 }
else if ((idade > 60) && (idade <= 64)) { pontos += 10 }
else if ((idade > 65) && (idade <= 69)) { pontos += 11 }
else if ((idade > 70) && (idade < 74)) { pontos += 12; }
else if ((idade > 75) && (idade <= 79)) { pontos += 13 }

```

```

if (colesterol < 160) { pontos += 0 }
else if (colesterol <= 199) {
    if (idade <= 39) { pontos += 4 }
    else if (idade <= 49) { pontos += 3 }
    else if (idade <= 59) { pontos += 2 }
    else if (idade <= 69) { pontos += 1 }
    else if (idade <= 79) { pontos += 0 }
}

```

```

else if (colesterol <= 239) {
    if (idade <= 39) { pontos += 7 }
    else if (idade <= 49) { pontos += 5 }
    else if (idade <= 59) { pontos += 3 }
    else if (idade <= 69) { pontos += 1 }
    else if (idade <= 79) { pontos += 0 }
}

```

```

else if (colesterol <= 279) {
    if (idade <= 39) { pontos += 9 }
    else if (idade <= 49) { pontos += 6 }
    else if (idade <= 59) { pontos += 4 }
    else if (idade <= 69) { pontos += 2 }
    else if (idade <= 79) { pontos += 1 }
}

```

```

else {
    if (idade <= 39) { pontos += 11 }
    else if (idade <= 49) { pontos += 8 }
    else if (idade <= 59) { pontos += 5 }
    else if (idade <= 69) { pontos += 3 }
    else if (idade <= 79) { pontos += 1 }
}

```

```

if (fumante.checked == true) {
    if (idade <= 39) { pontos += 8 }
    else if (idade <= 49) { pontos += 5 }
    else if (idade <= 59) { pontos += 3 }
    else if (idade <= 69) { pontos += 1 }
    else if (idade <= 79) { pontos += 1 }
}

```

```

if (pas2 < 120) { pontos += 0 }
else if (pas2 <= 129) {
    if (tratada.checked == true) { pontos += 1 }
    else { pontos += 0 }
}

```

```

else if (pas2 <= 139) {
    if (tratada.checked == true) { pontos += 2 }
    else { pontos += 1 }
}

```

```

else if (pas2 <= 159) {
    if (tratada.checked == true) { pontos += 2 }
    else { pontos += 1 }
}
else {
    if (tratada.checked == true) { pontos += 3 }
    else { pontos += 2 }
}
if (pontos < 0) { perc_risco = 0 }
else if (pontos <= 4) { perc_risco = 1 }
else if (pontos <= 6) { perc_risco = 2 }
else if (pontos == 7) { perc_risco = 3 }
else if (pontos == 8) { perc_risco = 4 }
else if (pontos == 9) { perc_risco = 5 }
else if (pontos == 10) { perc_risco = 6 }
else if (pontos == 11) { perc_risco = 8 }
else if (pontos == 12) { perc_risco = 10 }
else if (pontos == 13) { perc_risco = 12 }
else if (pontos == 14) { perc_risco = 16 }
else if (pontos == 15) { perc_risco = 20 }
else if (pontos == 16) { perc_risco = 25 }
else if (pontos >= 17) { perc_risco = 30 }

} else {
    if ((idade >= 20) && (idade <= 34)) { pontos += -7 }
    else if ((idade > 34) && (idade <= 39)) { pontos += -3 }
    else if ((idade > 40) && (idade <= 44)) { pontos += 0 }
    else if ((idade > 45) && (idade <= 49)) { pontos += 3 }
    else if ((idade > 50) && (idade <= 54)) { pontos += 6 }
    else if ((idade > 55) && (idade <= 59)) { pontos += 8 }
    else if ((idade > 60) && (idade <= 64)) { pontos += 10 }
    else if ((idade > 65) && (idade <= 69)) { pontos += 12 }
    else if ((idade > 70) && (idade < 74)) { pontos += 14 }
    else if ((idade > 75) && (idade <= 79)) { pontos += 16 }

    if (colesterol < 160) { pontos += 0 }
    else if (colesterol <= 199) {
        if (idade <= 39) { pontos += 4 }
        else if (idade <= 49) { pontos += 3 }
        else if (idade <= 59) { pontos += 2 }
        else if (idade <= 69) { pontos += 1 }
        else if (idade <= 79) { pontos += 1 }
    }
    else if (colesterol <= 239) {
        if (idade <= 39) { pontos += 8 }
        else if (idade <= 49) { pontos += 6 }
        else if (idade <= 59) { pontos += 4 }
        else if (idade <= 69) { pontos += 2 }
        else if (idade <= 79) { pontos += 1 }
    }
    else if (colesterol <= 279) {
        if (idade <= 39) { pontos += 11 }
        else if (idade <= 49) { pontos += 8 }
        else if (idade <= 59) { pontos += 5 }
        else if (idade <= 69) { pontos += 3 }
        else if (idade <= 79) { pontos += 2 }
    }
}

```

```

else {
    if (idade <= 39) { pontos += 13 }
    else if (idade <= 49) { pontos += 10 }
    else if (idade <= 59) { pontos += 7 }
    else if (idade <= 69) { pontos += 4 }
    else if (idade <= 79) { pontos += 2 }
}
if (fumante.checked == true) {
    if (idade <= 39) { pontos += 9 }
    else if (idade <= 49) { pontos += 7 }
    else if (idade <= 59) { pontos += 4 }
    else if (idade <= 69) { pontos += 2 }
    else if (idade <= 79) { pontos += 1 }
}

if (pas2 < 120) { pontos += 0 }
else if (pas2 <= 129) {
    if (tratada.checked == true) { pontos += 3 }
    else { pontos += 1 }
}
else if (pas2 <= 139) {
    if (tratada.checked == true) { pontos += 4 }
    else { pontos += 2 }
}
else if (pas2 <= 159) {
    if (tratada.checked == true) { pontos += 5 }
    else { pontos += 3 }
}
else {
    if (tratada.checked == true) { pontos += 6 }
    else { pontos += 4 }
}

if (pontos < 9) { perc_risco = 0 }
else if (pontos <= 12) { perc_risco = 1 }
else if (pontos <= 14) { perc_risco = 2 }
else if (pontos == 15) { perc_risco = 3 }
else if (pontos == 16) { perc_risco = 4 }
else if (pontos == 17) { perc_risco = 5 }
else if (pontos == 18) { perc_risco = 6 }
else if (pontos == 19) { perc_risco = 8 }
else if (pontos == 20) { perc_risco = 11 }
else if (pontos == 21) { perc_risco = 14 }
else if (pontos == 22) { perc_risco = 17 }
else if (pontos == 23) { perc_risco = 22 }
else if (pontos == 24) { perc_risco = 27 }
else if (pontos >= 25) { perc_risco = 30 }
}

console.log(pontos, idade);

if (perc_risco < 10) {
    retorno_calculo_estratificacao.innerHTML = 'BAIXO RISCO'
    localStorage.setItem('clas_estrat', 'BAIXO RISCO')
}

else if (perc_risco < 20) {
    retorno_calculo_estratificacao.innerHTML = 'RISCO MODERADO'
    localStorage.setItem('clas_estrat', 'RISCO MODERADO')
}

```

```

    }
    else {
        retorno_calculo_estratificacao.innerHTML = 'ALTO RISCO'
        localStorage.setItem('clas_estrat', 'ALTO RISCO')
    }
}
btnPasPad.addEventListener("click", calcular_pas_pad);
btnCirAbd.addEventListener("click", classif_circ_abdominal);
btnCintQuad.addEventListener("click", cint_quadril);
btnTornBraq.addEventListener("click", torno_brac);
btnEstrat2Calcular.addEventListener("click", calcularEstratificacaoParte2);

```

js/activeButton/activeButtonAva.js

```

const btn4 = document.querySelector('#btn-rastreamento');
const btn5 = document.querySelector('#btn-antropom');
const btn6 = document.querySelector('#btn-imc');
const btn7 = document.querySelector('#btn-circ-abdom');
const btn8 = document.querySelector('#btn-cint-quad');
const btn9 = document.querySelector('#btn-torn-brac');
const btn10 = document.querySelector('#btn-estrat-risco');

const tornBraqPasBra = document.querySelector('#torn-brac-pas-bra');

function carrega4(){
    const el = document.querySelector('.text-none-rastreamento');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn4.innerHTML = 'RASTREAMENTO DA HIPERTENSÃO ARTERIAL<span class="icon-expand"></span>';

    } else {
        el.style.display = 'block';
        btn4.textContent = 'FECHAR';
    }
}

function carrega5(){
    const el = document.querySelector('.text-none-antropom');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn5.innerHTML = 'DADOS ANTROPOMÉTRICOS<span class="icon-expand"></span>';

    } else {
        el.style.display = 'block';
        btn5.textContent = 'FECHAR';
    }
}

function carrega6(){
    const el = document.querySelector('.text-none-imc');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn6.innerHTML = 'ÍNDICE DE MASSA CORPORAL (IMC)<span class="icon-expand"></span>';

    } else {
        el.style.display = 'block';
    }
}

```

```

        btn6.textContent = 'FECHAR';
    }
}
function carrega7(){
    const el = document.querySelector('.text-none-circ-abdom');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn7.innerHTML = 'CIRCUNFERÊNCIA ABDOMINAL<span class="icon-expand"></span>';

        } else {
            el.style.display = 'block';
            btn7.textContent = 'FECHAR';
        }
    }
}
function carrega8(){
    const el = document.querySelector('.text-none-cint-quadr');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn8.innerHTML = 'RELAÇÃO CINTURA-QUADRIL<span class="icon-expand"></span>';

        } else {
            el.style.display = 'block';
            btn8.textContent = 'FECHAR';
        }
    }
}
function carrega9(){
    const el = document.querySelector('.text-none-torn-braq');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn9.innerHTML = 'ÍNDICE TORNOZELO-BRAQUIAL<span class="icon-expand"></span>';

        } else {
            el.style.display = 'block';
            btn9.textContent = 'FECHAR';
        }
    }
    //Retorna 'pas' informado no rastreamento
    //retornaPas();
}
function carrega10(){
    const el = document.querySelector('.text-none-estrat-risco');
    if(el.style.display === 'block'){
        el.style.display = 'none';
        btn10.innerHTML = 'ESTRATIFICAÇÃO DO RISCO CARDIOVASCULAR<span class="icon-
expand"></span>';

        } else {
            el.style.display = 'block';
            btn10.textContent = 'FECHAR';
        }
    }
}

function retornaPas(){
    tornBraqPasBra.value = localStorage.getItem('tornBraqPasBra');
}

```

```
}
```

```
btn4.addEventListener("click", carrega4);  
btn5.addEventListener("click", carrega5);  
btn6.addEventListener("click", carrega6);  
btn7.addEventListener("click", carrega7);  
btn8.addEventListener("click", carrega8);  
btn9.addEventListener("click", carrega9);  
btn10.addEventListener("click", carrega10);
```

```
js/activeButton/activeButtonGuiaEntrevista.js
```

```
const btn1 = document.querySelector('#btn-obj-entrevista');  
const btn2 = document.querySelector('#btn-quest-import');
```

```
function carrega1(){  
  const el = document.querySelector('.text-none-obj-entrevista');  
  if(el.style.display === 'block'){  
    el.style.display = 'none';  
    btn1.innerHTML = 'OBJETIVOS PARA ENTREVISTA<span class="icon-expand"></span>';  
  
    } else {  
      el.style.display = 'block';  
      btn1.textContent = 'FECHAR';  
    }  
  }  
}  
function carrega2(){  
  const el = document.querySelector('.text-none-quest-import');  
  if(el.style.display === 'block'){  
    el.style.display = 'none';  
    btn2.innerHTML = 'QUESTIONAMENTOS IMPORTANTES<span class="icon-expand"></span>';  
  
    } else {  
      el.style.display = 'block';  
      btn2.textContent = 'FECHAR';  
    }  
  }  
}
```

```
btn1.addEventListener("click", carrega1);  
btn2.addEventListener("click", carrega2);
```

```
js/activeButton/activeButtonRecom.js
```

```
const btn1 = document.querySelector('#btn-aspectos');  
const btn2 = document.querySelector('#btn-proped');  
const btn3 = document.querySelector('#btn-sinais');
```

```
function carrega1(){  
  const el = document.querySelector('.text-none-aspectos');  
  if(el.style.display === 'block'){  
    el.style.display = 'none';  
    btn1.innerHTML = 'ASPECTOS A SEREM AVALIADOS<span class="icon-expand"></span>';
```

```

    } else {
      el.style.display = 'block';
      btn1.textContent = 'FECHAR';
    }
  }
function carrega2(){
  const el = document.querySelector('.text-none-proped');
  if(el.style.display === 'block'){
    el.style.display = 'none';
    btn2.innerHTML = 'TÉCNICAS PROPEDEÚTICAS<span class="icon-expand"></span>';

    } else {
      el.style.display = 'block';
      btn2.textContent = 'FECHAR';
    }
  }
function carrega3(){
  const el = document.querySelector('.text-none-sinais');
  if(el.style.display === 'block'){
    el.style.display = 'none';
    btn3.innerHTML = 'SINAIS VITAIS<span class="icon-expand"></span>';

    } else {
      el.style.display = 'block';
      btn3.textContent = 'FECHAR';
    }
  }
}

btn1.addEventListener("click", carrega1);
btn2.addEventListener("click", carrega2);
btn3.addEventListener("click", carrega3);

```

js/bootstrap-js/bootstrap.bundle.js

```

/#!
* Bootstrap v5.2.2 (https://getbootstrap.com/)
* Copyright 2011-2022 The Bootstrap Authors
(https://github.com/twbs/bootstrap/graphs/contributors)
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
*/
(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? module.exports = factory() :
  typeof define === 'function' && define.amd ? define(factory) :
  (global = typeof globalThis !== 'undefined' ? globalThis : global || self, global.bootstrap = factory());
})(this, (function () { 'use strict';

  /**
   * -----
   * Bootstrap (v5.2.2): util/index.js
   * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
   * -----
   */
  const MAX_UID = 1000000;
  const MILLISECONDS_MULTIPLIER = 1000;
  const TRANSITION_END = 'transitionend'; // Shout-out Angus Croll (https://goo.gl/pxwQGp)

```



```

const toType = object => {
  if (object === null || object === undefined) {
    return `${object}`;
  }

  return Object.prototype.toString.call(object).match(/\s([a-z]+)/i)[1].toLowerCase();
};
/**
 * Public Util API
 */

const getUID = prefix => {
  do {
    prefix += Math.floor(Math.random() * MAX_UID);
  } while (document.getElementById(prefix));

  return prefix;
};

const getSelector = element => {
  let selector = element.getAttribute('data-bs-target');

  if (!selector || selector === '#') {
    let hrefAttribute = element.getAttribute('href'); // The only valid content that could double as a
    selector are IDs or classes,
    // so everything starting with `#` or `.`. If a "real" URL is used as the selector,
    // `document.querySelector` will rightfully complain it is invalid.
    // See https://github.com/twbs/bootstrap/issues/32273

    if (!hrefAttribute || !hrefAttribute.includes('#') && !hrefAttribute.startsWith('.')) {
      return null;
    } // Just in case some CMS puts out a full URL with the anchor appended

    if (hrefAttribute.includes('#') && !hrefAttribute.startsWith('#')) {
      hrefAttribute = `#${hrefAttribute.split('#')[1]}`;
    }

    selector = hrefAttribute && hrefAttribute !== '#' ? hrefAttribute.trim() : null;
  }

  return selector;
};

const getSelectorFromElement = element => {
  const selector = getSelector(element);

  if (selector) {
    return document.querySelector(selector) ? selector : null;
  }

  return null;
};

const getElementFromSelector = element => {

```

```

const selector = getSelector(element);
return selector ? document.querySelector(selector) : null;
};

const getTransitionDurationFromElement = element => {
  if (!element) {
    return 0;
  } // Get transition-duration of the element

  let {
    transitionDuration,
    transitionDelay
  } = window.getComputedStyle(element);
  const floatTransitionDuration = Number.parseFloat(transitionDuration);
  const floatTransitionDelay = Number.parseFloat(transitionDelay); // Return 0 if element or
  transition duration is not found

  if (!floatTransitionDuration && !floatTransitionDelay) {
    return 0;
  } // If multiple durations are defined, take the first

  transitionDuration = transitionDuration.split(',')[0];
  transitionDelay = transitionDelay.split(',')[0];
  return (Number.parseFloat(transitionDuration) + Number.parseFloat(transitionDelay)) *
  MILLISECONDS_MULTIPLIER;
};

const triggerTransitionEnd = element => {
  element.dispatchEvent(new Event(TRANSITION_END));
};

const isElement$1 = object => {
  if (!object || typeof object !== 'object') {
    return false;
  }

  if (typeof object.jquery !== 'undefined') {
    object = object[0];
  }

  return typeof object.nodeType !== 'undefined';
};

const getElement = object => {
  // it's a jQuery object or a node element
  if (isElement$1(object)) {
    return object.jquery ? object[0] : object;
  }

  if (typeof object === 'string' && object.length > 0) {
    return document.querySelector(object);
  }

  return null;
};

```

```

const isVisible = element => {
  if (!isElement$1(element) || element.getClientRects().length === 0) {
    return false;
  }

  const elementIsVisible = getComputedStyle(element).getPropertyValue('visibility') === 'visible'; //
  Handle `details` element as its content may false appear visible when it is closed

  const closedDetails = element.closest('details:not([open])');

  if (!closedDetails) {
    return elementIsVisible;
  }

  if (closedDetails !== element) {
    const summary = element.closest('summary');

    if (summary && summary.parentNode !== closedDetails) {
      return false;
    }

    if (summary === null) {
      return false;
    }
  }

  return elementIsVisible;
};

const isDisabled = element => {
  if (!element || element.nodeType !== Node.ELEMENT_NODE) {
    return true;
  }

  if (element.classList.contains('disabled')) {
    return true;
  }

  if (typeof element.disabled !== 'undefined') {
    return element.disabled;
  }

  return element.hasAttribute('disabled') && element.getAttribute('disabled') !== 'false';
};

const findShadowRoot = element => {
  if (!document.documentElement.attachShadow) {
    return null;
  }
  // Can find the shadow root otherwise it'll return the document

  if (typeof element.getRootNode === 'function') {
    const root = element.getRootNode();
    return root instanceof ShadowRoot ? root : null;
  }

```

```

    if (element instanceof ShadowRoot) {
        return element;
    } // when we don't find a shadow root

    if (!element.parentNode) {
        return null;
    }

    return findShadowRoot(element.parentNode);
};

const noop = () => {};
/**
 * Trick to restart an element's animation
 *
 * @param {HTMLElement} element
 * @return void
 *
 * @see https://www.charistheo.io/blog/2021/02/restart-a-css-animation-with-javascript/#restarting-a-css-animation
 */

const reflow = element => {
    element.offsetHeight; // eslint-disable-line no-unused-expressions
};

const getjQuery = () => {
    if (window.jQuery && !document.body.hasAttribute('data-bs-no-jquery')) {
        return window.jQuery;
    }

    return null;
};

const DOMContentLoadedCallbacks = [];

const onDOMContentLoaded = callback => {
    if (document.readyState === 'loading') {
        // add listener on the first call when the document is in loading state
        if (!DOMContentLoadedCallbacks.length) {
            document.addEventListener('DOMContentLoaded', () => {
                for (const callback of DOMContentLoadedCallbacks) {
                    callback();
                }
            });
        }
    }

    DOMContentLoadedCallbacks.push(callback);
    if (callback) {
        callback();
    }
};

const isRTL = () => document.documentElement.dir === 'rtl';

```

```

const defineJQueryPlugin = plugin => {
  onDOMContentLoaded(() => {
    const $ = getjQuery();
    /* istanbul ignore if */

    if ($) {
      const name = plugin.NAME;
      const JQUERY_NO_CONFLICT = $.fn[name];
      $.fn[name] = plugin.jQueryInterface;
      $.fn[name].Constructor = plugin;

      $.fn[name].noConflict = () => {
        $.fn[name] = JQUERY_NO_CONFLICT;
        return plugin.jQueryInterface;
      };
    }
  });
};

const execute = callback => {
  if (typeof callback === 'function') {
    callback();
  }
};

const executeAfterTransition = (callback, transitionElement, waitForTransition = true) => {
  if (!waitForTransition) {
    execute(callback);
    return;
  }

  const durationPadding = 5;
  const emulatedDuration = getTransitionDurationFromElement(transitionElement) +
    durationPadding;
  let called = false;

  const handler = ({
    target
  }) => {
    if (target !== transitionElement) {
      return;
    }

    called = true;
    transitionElement.removeEventListener(TRANSITION_END, handler);
    execute(callback);
  };

  transitionElement.addEventListener(TRANSITION_END, handler);
  setTimeout(() => {
    if (!called) {
      triggerTransitionEnd(transitionElement);
    }
  }, emulatedDuration);
};
/**
 * Return the previous/next element of a list.

```

```

*
* @param {array} list The list of elements
* @param activeElement The active element
* @param shouldGetNext Choose to get next or previous element
* @param isCycleAllowed
* @return {Element|elem} The proper element
*/

const getNextActiveElement = (list, activeElement, shouldGetNext, isCycleAllowed) => {
  const listLength = list.length;
  let index = list.indexOf(activeElement); // if the element does not exist in the list return an element
  // depending on the direction and if cycle is allowed

  if (index === -1) {
    return !shouldGetNext && isCycleAllowed ? list[listLength - 1] : list[0];
  }

  index += shouldGetNext ? 1 : -1;

  if (isCycleAllowed) {
    index = (index + listLength) % listLength;
  }

  return list[Math.max(0, Math.min(index, listLength - 1))];
};

/**
 * -----
 * Bootstrap (v5.2.2): dom/event-handler.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const namespaceRegex = /[\^.]*(?=\..*)\.\.*/;
const stripNameRegex = /\..*/;
const stripUidRegex = /::\d+$/;
const eventRegistry = {}; // Events storage

let uidEvent = 1;
const customEvents = {
  mouseenter: 'mouseover',
  mouseleave: 'mouseout'
};
const nativeEvents = new Set(['click', 'dblclick', 'mouseup', 'mousedown', 'contextmenu', 'mousewheel', 'DOMMouseScroll', 'mouseover', 'mouseout', 'mousemove', 'selectstart', 'selectend', 'keydown', 'keypress', 'keyup', 'orientationchange', 'touchstart', 'touchmove', 'touchend', 'touchcancel', 'pointerdown', 'pointermove', 'pointerup', 'pointerleave', 'pointercancel', 'gesturestart', 'gesturechange', 'gestureend', 'focus', 'blur', 'change', 'reset', 'select', 'submit', 'focusin', 'focusout', 'load', 'unload', 'beforeunload', 'resize', 'move', 'DOMContentLoaded', 'readystatechange', 'error', 'abort', 'scroll']);
/**
 * Private methods
 */

```

```

function makeEventUid(element, uid) {
  return uid && `${uid}::${uidEvent++}` || element.uidEvent || uidEvent++;
}

function getElementEvents(element) {
  const uid = makeEventUid(element);
  element.uidEvent = uid;
  eventRegistry[uid] = eventRegistry[uid] || {};
  return eventRegistry[uid];
}

function bootstrapHandler(element, fn) {
  return function handler(event) {
    hydrateObj(event, {
      delegateTarget: element
    });

    if (handler.oneOff) {
      EventHandler.off(element, event.type, fn);
    }

    return fn.apply(element, [event]);
  };
}

function bootstrapDelegationHandler(element, selector, fn) {
  return function handler(event) {
    const domElements = element.querySelectorAll(selector);

    for (let {
      target
    } = event; target && target !== this; target = target.parentNode) {
      for (const domElement of domElements) {
        if (domElement !== target) {
          continue;
        }

        hydrateObj(event, {
          delegateTarget: target
        });

        if (handler.oneOff) {
          EventHandler.off(element, event.type, selector, fn);
        }

        return fn.apply(target, [event]);
      }
    }
  };
}

function findHandler(events, callable, delegationSelector = null) {
  return Object.values(events).find(event => event.callable === callable && event.delegationSelector === delegationSelector);
}

```

```

function normalizeParameters(originalTypeEvent, handler, delegationFunction) {
  const isDelegated = typeof handler === 'string'; // todo: tooltip passes `false` instead of selector, so
  we need to check

  const callable = isDelegated ? delegationFunction : handler || delegationFunction;
  let typeEvent = getTypeEvent(originalTypeEvent);

  if (!nativeEvents.has(typeEvent)) {
    typeEvent = originalTypeEvent;
  }

  return [isDelegated, callable, typeEvent];
}

function addHandler(element, originalTypeEvent, handler, delegationFunction, oneOff) {
  if (typeof originalTypeEvent !== 'string' || !element) {
    return;
  }

  let [isDelegated, callable, typeEvent] = normalizeParameters(originalTypeEvent, handler,
    delegationFunction); // in case of mouseenter or mouseleave wrap the handler within a function that
    checks for its DOM position
    // this prevents the handler from being dispatched the same way as mouseover or mouseout does

  if (originalTypeEvent in customEvents) {
    const wrapFunction = fn => {
      return function (event) {
        if (!event.relatedTarget || event.relatedTarget !== event.delegateTarget
          && !event.delegateTarget.contains(event.relatedTarget)) {
          return fn.call(this, event);
        }
      };
    };

    callable = wrapFunction(callable);
  }

  const events = getElementEvents(element);
  const handlers = events[typeEvent] || (events[typeEvent] = {});
  const previousFunction = findHandler(handlers, callable, isDelegated ? handler : null);

  if (previousFunction) {
    previousFunction.oneOff = previousFunction.oneOff && oneOff;
    return;
  }

  const uid = makeEventUid(callable, originalTypeEvent.replace(namespaceRegex, ''));
  const fn = isDelegated ? bootstrapDelegationHandler(element, handler, callable) :
  bootstrapHandler(element, callable);
  fn.delegationSelector = isDelegated ? handler : null;
  fn.callable = callable;
  fn.oneOff = oneOff;
  fn.uidEvent = uid;
  handlers[uid] = fn;
  element.addEventListener(typeEvent, fn, isDelegated);
}

```



```

function removeHandler(element, events, typeEvent, handler, delegationSelector) {
  const fn = findHandler(events[typeEvent], handler, delegationSelector);

  if (!fn) {
    return;
  }

  element.removeEventListener(typeEvent, fn, Boolean(delegationSelector));
  delete events[typeEvent][fn.uidEvent];
}

function removeNamespacedHandlers(element, events, typeEvent, namespace) {
  const storeElementEvent = events[typeEvent] || {};

  for (const handlerKey of Object.keys(storeElementEvent)) {
    if (handlerKey.includes(namespace)) {
      const event = storeElementEvent[handlerKey];
      removeHandler(element, events, typeEvent, event.callable, event.delegationSelector);
    }
  }
}

function getTypeEvent(event) {
  // allow to get the native events from namespaced events ('click.bs.button' --> 'click')
  event = event.replace(stripNameRegex, "");
  return customEvents[event] || event;
}

const EventHandler = {
  on(element, event, handler, delegationFunction) {
    addHandler(element, event, handler, delegationFunction, false);
  },

  one(element, event, handler, delegationFunction) {
    addHandler(element, event, handler, delegationFunction, true);
  },

  off(element, originalTypeEvent, handler, delegationFunction) {
    if (typeof originalTypeEvent !== 'string' || !element) {
      return;
    }

    const [isDelegated, callable, typeEvent] = normalizeParameters(originalTypeEvent, handler,
    delegationFunction);
    const inNamespace = typeEvent !== originalTypeEvent;
    const events = getElementEvents(element);
    const storeElementEvent = events[typeEvent] || {};
    const isNamespace = originalTypeEvent.startsWith('.');

    if (typeof callable !== 'undefined') {
      // Simplest case: handler is passed, remove that listener ONLY.
      if (!Object.keys(storeElementEvent).length) {
        return;
      }
    }

    removeHandler(element, events, typeEvent, callable, isDelegated ? handler : null);
    return;
  }
};

```

```

    }

    if (isNamespace) {
      for (const elementEvent of Object.keys(events)) {
        removeNamespacedHandlers(element, events, elementEvent, originalTypeEvent.slice(1));
      }
    }

    for (const keyHandlers of Object.keys(storeElementEvent)) {
      const handlerKey = keyHandlers.replace(stripUidRegex, "");

      if (!inNamespace || originalTypeEvent.includes(handlerKey)) {
        const event = storeElementEvent[keyHandlers];
        removeHandler(element, events, typeEvent, event.callable, event.delegationSelector);
      }
    }
  },

  trigger(element, event, args) {
    if (typeof event !== 'string' || !element) {
      return null;
    }

    const $ = getjQuery();
    const typeEvent = getTypeEvent(event);
    const inNamespace = event !== typeEvent;
    let jQueryEvent = null;
    let bubbles = true;
    let nativeDispatch = true;
    let defaultPrevented = false;

    if (inNamespace && $) {
      jQueryEvent = $.Event(event, args);
      $(element).trigger(jQueryEvent);
      bubbles = !jQueryEvent.isPropagationStopped();
      nativeDispatch = !jQueryEvent.isImmediatePropagationStopped();
      defaultPrevented = jQueryEvent.isDefaultPrevented();
    }

    let evt = new Event(event, {
      bubbles,
      cancelable: true
    });
    evt = hydrateObj(evt, args);

    if (defaultPrevented) {
      evt.preventDefault();
    }

    if (nativeDispatch) {
      element.dispatchEvent(evt);
    }

    if (evt.defaultPrevented && jQueryEvent) {
      jQueryEvent.preventDefault();
    }
  }

```

```

    return evt;
  }

};

function hydrateObj(obj, meta) {
  for (const [key, value] of Object.entries(meta || {})) {
    try {
      obj[key] = value;
    } catch (_unused) {
      Object.defineProperty(obj, key, {
        configurable: true,

        get() {
          return value;
        }

      });
    }
  }

  return obj;
}

/**
 * -----
 * Bootstrap (v5.2.2): dom/data.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */

/**
 * Constants
 */
const elementMap = new Map();
const Data = {
  set(element, key, instance) {
    if (!elementMap.has(element)) {
      elementMap.set(element, new Map());
    }

    const instanceMap = elementMap.get(element); // make it clear we only want one instance per element
    // can be removed later when multiple key/instances are fine to be used

    if (!instanceMap.has(key) && instanceMap.size !== 0) {
      // eslint-disable-next-line no-console
      console.error(`Bootstrap doesn't allow more than one instance per element. Bound instance: ${Array.from(instanceMap.keys())[0]}.`);
      return;
    }

    instanceMap.set(key, instance);
  },

  get(element, key) {
    if (elementMap.has(element)) {

```

```

        return elementMap.get(element).get(key) || null;
    }

    return null;
},

remove(element, key) {
    if (!elementMap.has(element)) {
        return;
    }

    const instanceMap = elementMap.get(element);
    instanceMap.delete(key); // free up element references if there are no instances left for an
    element

    if (instanceMap.size === 0) {
        elementMap.delete(element);
    }
}

};

/**
 * -----
 * Bootstrap (v5.2.2): dom/manipulator.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
function normalizeData(value) {
    if (value === 'true') {
        return true;
    }

    if (value === 'false') {
        return false;
    }

    if (value === Number(value).toString()) {
        return Number(value);
    }

    if (value === '' || value === 'null') {
        return null;
    }

    if (typeof value !== 'string') {
        return value;
    }

    try {
        return JSON.parse(decodeURIComponent(value));
    } catch (_unused) {
        return value;
    }
}

function normalizeDataKey(key) {

```

```

    return key.replace(/[A-Z]/g, chr => `-${chr.toLowerCase()}`);
}

const Manipulator = {
  setDataAttribute(element, key, value) {
    element.setAttribute(`data-bs-${normalizeDataKey(key)}`, value);
  },

  removeDataAttribute(element, key) {
    element.removeAttribute(`data-bs-${normalizeDataKey(key)}`);
  },

  getDataAttributes(element) {
    if (!element) {
      return {};
    }

    const attributes = {};
    const bsKeys = Object.keys(element.dataset).filter(key => key.startsWith('bs')
    && !key.startsWith('bsConfig'));

    for (const key of bsKeys) {
      let pureKey = key.replace(/^bs/, "");
      pureKey = pureKey.charAt(0).toLowerCase() + pureKey.slice(1, pureKey.length);
      attributes[pureKey] = normalizeData(element.dataset[key]);
    }

    return attributes;
  },

  getDataAttribute(element, key) {
    return normalizeData(element.getAttribute(`data-bs-${normalizeDataKey(key)}`));
  }
};

/**
 * -----
 * Bootstrap (v5.2.2): util/config.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Class definition
 */

class Config {
  // Getters
  static get Default() {
    return {};
  }

  static get DefaultType() {
    return {};
  }

  static get NAME() {

```

```

    throw new Error('You have to implement the static method "NAME", for each component!');
  }

  _getConfig(config) {
    config = this._mergeConfigObj(config);
    config = this._configAfterMerge(config);

    this._typeCheckConfig(config);

    return config;
  }

  _configAfterMerge(config) {
    return config;
  }

  _mergeConfigObj(config, element) {
    const jsonConfig = isElement$1(element) ? Manipulator.getDataAttribute(element, 'config') : {}; //
    try to parse

    return { ...this.constructor.Default,
      ...(typeof jsonConfig === 'object' ? jsonConfig : {}),
      ...(isElement$1(element) ? Manipulator.getDataAttributes(element) : {}),
      ...(typeof config === 'object' ? config : {})};
  }

  _typeCheckConfig(config, configTypes = this.constructor.DefaultType) {
    for (const property of Object.keys(configTypes)) {
      const expectedTypes = configTypes[property];
      const value = config[property];
      const valueType = isElement$1(value) ? 'element' : toType(value);

      if (!new RegExp(expectedTypes).test(valueType)) {
        throw new TypeError(`${this.constructor.NAME.toUpperCase()}: Option "${property}" provided
        type "${valueType}" but expected type "${expectedTypes}".`);
      }
    }
  }
}

/**
 * -----
 * Bootstrap (v5.2.2): base-component.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const VERSION = '5.2.2';
/**
 * Class definition
 */

```

```

class BaseComponent extends Config {
  constructor(element, config) {
    super();
    element = getElement(element);

    if (!element) {
      return;
    }

    this._element = element;
    this._config = this._getConfig(config);
    Data.set(this._element, this.constructor.DATA_KEY, this);
  } // Public

  dispose() {
    Data.remove(this._element, this.constructor.DATA_KEY);
    EventHandler.off(this._element, this.constructor.EVENT_KEY);

    for (const propertyName of Object.getOwnPropertyNames(this)) {
      this[propertyName] = null;
    }
  }

  _queueCallback(callback, element, isAnimated = true) {
    executeAfterTransition(callback, element, isAnimated);
  }

  _getConfig(config) {
    config = this._mergeConfigObj(config, this._element);
    config = this._configAfterMerge(config);

    this._typeCheckConfig(config);

    return config;
  } // Static

  static getInstance(element) {
    return Data.get(getElement(element), this.DATA_KEY);
  }

  static getOrCreateInstance(element, config = {}) {
    return this.getInstance(element) || new this(element, typeof config === 'object' ? config : null);
  }

  static get VERSION() {
    return VERSION;
  }

  static get DATA_KEY() {
    return `bs.${this.NAME}`;
  }

  static get EVENT_KEY() {
    return `.${this.DATA_KEY}`;
  }
}

```

```

static eventName(name) {
  return `${name}${this.EVENT_KEY}`;
}

}

/**
 * -----
 * Bootstrap (v5.2.2): util/component-functions.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */

const enableDismissTrigger = (component, method = 'hide') => {
  const clickEvent = `click.dismiss${component.EVENT_KEY}`;
  const name = component.NAME;
  EventHandler.on(document, clickEvent, `[data-bs-dismiss="${name}"]`, function (event) {
    if (['A', 'AREA'].includes(this.tagName)) {
      event.preventDefault();
    }

    if (isDisabled(this)) {
      return;
    }

    const target = getElementFromSelector(this) || this.closest(`.${name}`);
    const instance = component.getOrCreateInstance(target); // Method argument is left, for Alert
    and only, as it doesn't implement the 'hide' method

    instance[method]();
  });
};

/**
 * -----
 * Bootstrap (v5.2.2): alert.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */

/**
 * Constants
 */

const NAME$f = 'alert';
const DATA_KEY$a = 'bs.alert';
const EVENT_KEY$b = `.${DATA_KEY$a}`;
const EVENT_CLOSE = `close${EVENT_KEY$b}`;
const EVENT_CLOSED = `closed${EVENT_KEY$b}`;
const CLASS_NAME_FADE$5 = 'fade';
const CLASS_NAME_SHOW$8 = 'show';

/**
 * Class definition
 */

class Alert extends BaseComponent {
  // Getters

```



```

static get NAME() {
    return NAME$f;
} // Public

close() {
    const closeEvent = EventHandler.trigger(this._element, EVENT_CLOSE);

    if (closeEvent.defaultPrevented) {
        return;
    }

    this._element.classList.remove(CLASS_NAME_SHOW$8);

    const isAnimated = this._element.classList.contains(CLASS_NAME_FADE$5);

    this._queueCallback(() => this._destroyElement(), this._element, isAnimated);
} // Private

_destroyElement() {
    this._element.remove();

    EventHandler.trigger(this._element, EVENT_CLOSED);
    this.dispose();
} // Static

static jQueryInterface(config) {
    return this.each(function () {
        const data = Alert.getOrCreateInstance(this);

        if (typeof config !== 'string') {
            return;
        }

        if (data[config] === undefined || config.startsWith('_') || config === 'constructor') {
            throw new TypeError(`No method named "${config}"`);
        }

        data[config](this);
    });
}

/**
 * Data API implementation
 */

enableDismissTrigger(Alert, 'close');

/**
 * jQuery
 */

defineJQueryPlugin(Alert);

```

```

/**
 * -----
 * Bootstrap (v5.2.2): button.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$e = 'button';
const DATA_KEY$9 = 'bs.button';
const EVENT_KEY$a = `.${DATA_KEY$9}`;
const DATA_API_KEY$6 = '.data-api';
const CLASS_NAME_ACTIVE$3 = 'active';
const SELECTOR_DATA_TOGGLE$5 = '[data-bs-toggle="button"]';
const EVENT_CLICK_DATA_API$6 = `click${EVENT_KEY$a}${DATA_API_KEY$6}`;
/**
 * Class definition
 */

class Button extends BaseComponent {
  // Getters
  static get NAME() {
    return NAME$e;
  } // Public

  toggle() {
    // Toggle class and sync the `aria-pressed` attribute with the return value of the `.toggle()` method
    this._element.setAttribute('aria-pressed', this._element.classList.toggle(CLASS_NAME_ACTIVE$3));
  } // Static

  static jQueryInterface(config) {
    return this.each(function () {
      const data = Button.getOrCreateInstance(this);

      if (config === 'toggle') {
        data[config]();
      }
    });
  }
}
/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API$6, SELECTOR_DATA_TOGGLE$5, event => {
  event.preventDefault();
  const button = event.target.closest(SELECTOR_DATA_TOGGLE$5);
  const data = Button.getOrCreateInstance(button);
  data.toggle();
});
/**

```

```

* jQuery
*/

defineJQueryPlugin(Button);

/**
 * -----
 * Bootstrap (v5.2.2): dom/selector-engine.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const SelectorEngine = {
  find(selector, element = document.documentElement) {
    return [].concat(...Element.prototype.querySelectorAll.call(element, selector));
  },

  findOne(selector, element = document.documentElement) {
    return Element.prototype.querySelector.call(element, selector);
  },

  children(element, selector) {
    return [].concat(...element.children).filter(child => child.matches(selector));
  },

  parents(element, selector) {
    const parents = [];
    let ancestor = element.parentNode.closest(selector);

    while (ancestor) {
      parents.push(ancestor);
      ancestor = ancestor.parentNode.closest(selector);
    }

    return parents;
  },

  prev(element, selector) {
    let previous = element.previousElementSibling;

    while (previous) {
      if (previous.matches(selector)) {
        return [previous];
      }

      previous = previous.previousElementSibling;
    }

    return [];
  },

  // TODO: this is now unused; remove later along with prev()
  next(element, selector) {
    let next = element.nextElementSibling;

```

```

while (next) {
  if (next.matches(selector)) {
    return [next];
  }

  next = next.nextElementSibling;
}

return [];
},

focusableChildren(element) {
  const focusables = ['a', 'button', 'input', 'textarea', 'select', 'details', '[tabindex]',
'contenteditable="true"'].map(selector => `${selector}:not([tabindex^="-"])`).join(',');
  return this.find(focusables, element).filter(el => !isDisabled(el) && isVisible(el));
}

};

/**
 * -----
 * Bootstrap (v5.2.2): util/swipe.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$d = 'swipe';
const EVENT_KEY$9 = '.bs.swipe';
const EVENT_TOUCHSTART = `touchstart${EVENT_KEY$9}`;
const EVENT_TOUCHMOVE = `touchmove${EVENT_KEY$9}`;
const EVENT_TOUCHEND = `touchend${EVENT_KEY$9}`;
const EVENT_POINTERDOWN = `pointerdown${EVENT_KEY$9}`;
const EVENT_POINTERUP = `pointerup${EVENT_KEY$9}`;
const POINTER_TYPE_TOUCH = 'touch';
const POINTER_TYPE_PEN = 'pen';
const CLASS_NAME_POINTER_EVENT = 'pointer-event';
const SWIPE_THRESHOLD = 40;
const Default$c = {
  endCallback: null,
  leftCallback: null,
  rightCallback: null
};
const DefaultType$c = {
  endCallback: '(function|null)',
  leftCallback: '(function|null)',
  rightCallback: '(function|null)'
};
/**
 * Class definition
 */

class Swipe extends Config {
  constructor(element, config) {

```

```

super();
this._element = element;

if (!element || !Swipe.isSupported()) {
    return;
}

this._config = this._getConfig(config);
this._deltaX = 0;
this._supportPointerEvents = Boolean(window.PointerEvent);

this._initEvents();
} // Getters

static get Default() {
    return Default$c;
}

static get DefaultType() {
    return DefaultType$c;
}

static get NAME() {
    return NAME$d;
} // Public

dispose() {
    EventHandler.off(this._element, EVENT_KEY$9);
} // Private

_start(event) {
    if (!this._supportPointerEvents) {
        this._deltaX = event.touches[0].clientX;
        return;
    }

    if (this._eventIsPointerPenTouch(event)) {
        this._deltaX = event.clientX;
    }
}

_end(event) {
    if (this._eventIsPointerPenTouch(event)) {
        this._deltaX = event.clientX - this._deltaX;
    }
}

this._handleSwipe();

execute(this._config.endCallback);
}

_move(event) {
    this._deltaX = event.touches && event.touches.length > 1 ? 0 : event.touches[0].clientX -
this._deltaX;

```

```

    }

    _handleSwipe() {
    const absDeltaX = Math.abs(this._deltaX);

    if (absDeltaX <= SWIPE_THRESHOLD) {
        return;
    }

    const direction = absDeltaX / this._deltaX;
    this._deltaX = 0;

    if (!direction) {
        return;
    }

    execute(direction > 0 ? this._config.rightCallback : this._config.leftCallback);
}

_initEvents() {
    if (this._supportPointerEvents) {
        EventHandler.on(this._element, EVENT_POINTERDOWN, event => this._start(event));
        EventHandler.on(this._element, EVENT_POINTERUP, event => this._end(event));

        this._element.classList.add(CLASS_NAME_POINTER_EVENT);
    } else {
        EventHandler.on(this._element, EVENT_TOUCHSTART, event => this._start(event));
        EventHandler.on(this._element, EVENT_TOUCHMOVE, event => this._move(event));
        EventHandler.on(this._element, EVENT_TOUCHEND, event => this._end(event));
    }
}

_eventIsPointerPenTouch(event) {
    return this._supportPointerEvents && (event.pointerType === POINTER_TYPE_PEN ||
event.pointerType === POINTER_TYPE_TOUCH);
} // Static

static isSupported() {
    return 'ontouchstart' in document.documentElement || navigator.maxTouchPoints > 0;
}

}

/**
 * -----
 * Bootstrap (v5.2.2): carousel.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$c = 'carousel';
const DATA_KEY$8 = 'bs.carousel';
const EVENT_KEY$8 = `.${DATA_KEY$8}`;

```

```

const DATA_API_KEY$5 = '.data-api';
const ARROW_LEFT_KEY$1 = 'ArrowLeft';
const ARROW_RIGHT_KEY$1 = 'ArrowRight';
const TOUCHEVENT_COMPAT_WAIT = 500; // Time for mouse compat events to fire after touch

```

```

const ORDER_NEXT = 'next';
const ORDER_PREV = 'prev';
const DIRECTION_LEFT = 'left';
const DIRECTION_RIGHT = 'right';
const EVENT_SLIDE = `slide${EVENT_KEY$8}`;
const EVENT_SLID = `slid${EVENT_KEY$8}`;
const EVENT_KEYDOWN$1 = `keydown${EVENT_KEY$8}`;
const EVENT_MOUSEENTER$1 = `mouseenter${EVENT_KEY$8}`;
const EVENT_MOUSELEAVE$1 = `mouseleave${EVENT_KEY$8}`;
const EVENT_DRAG_START = `dragstart${EVENT_KEY$8}`;
const EVENT_LOAD_DATA_API$3 = `load${EVENT_KEY$8}${DATA_API_KEY$5}`;
const EVENT_CLICK_DATA_API$5 = `click${EVENT_KEY$8}${DATA_API_KEY$5}`;
const CLASS_NAME_CAROUSEL = 'carousel';
const CLASS_NAME_ACTIVE$2 = 'active';
const CLASS_NAME_SLIDE = 'slide';
const CLASS_NAME_END = 'carousel-item-end';
const CLASS_NAME_START = 'carousel-item-start';
const CLASS_NAME_NEXT = 'carousel-item-next';
const CLASS_NAME_PREV = 'carousel-item-prev';
const SELECTOR_ACTIVE = '.active';
const SELECTOR_ITEM = '.carousel-item';
const SELECTOR_ACTIVE_ITEM = SELECTOR_ACTIVE + SELECTOR_ITEM;
const SELECTOR_ITEM_IMG = '.carousel-item img';
const SELECTOR_INDICATORS = '.carousel-indicators';
const SELECTOR_DATA_SLIDE = '[data-bs-slide], [data-bs-slide-to]';
const SELECTOR_DATA_RIDE = '[data-bs-ride="carousel"]';
const KEY_TO_DIRECTION = {
  [ARROW_LEFT_KEY$1]: DIRECTION_RIGHT,
  [ARROW_RIGHT_KEY$1]: DIRECTION_LEFT
};
const Default$b = {
  interval: 5000,
  keyboard: true,
  pause: 'hover',
  ride: false,
  touch: true,
  wrap: true
};
const DefaultType$b = {
  interval: '(number|boolean)',
  // TODO:v6 remove boolean support
  keyboard: 'boolean',
  pause: '(string|boolean)',
  ride: '(boolean|string)',
  touch: 'boolean',
  wrap: 'boolean'
};
/**
 * Class definition
 */

```

```

class Carousel extends BaseComponent {

```

```

constructor(element, config) {
    super(element, config);
    this._interval = null;
    this._activeElement = null;
    this._isSliding = false;
    this.touchTimeout = null;
    this._swipeHelper = null;
    this._indicatorsElement = SelectorEngine.findOne(SELECTOR_INDICATORS, this._element);

    this._addEventListeners();

    if (this._config.ride === CLASS_NAME_CAROUSEL) {
        this.cycle();
    }
} // Getters

static get Default() {
    return Default$b;
}

static get DefaultType() {
    return DefaultType$b;
}

static get NAME() {
    return NAME$c;
} // Public

next() {
    this._slide(ORDER_NEXT);
}

nextWhenVisible() {
    // FIXME TODO use `document.visibilityState`
    // Don't call next when the page isn't visible
    // or the carousel or its parent isn't visible
    if (!document.hidden && isVisible(this._element)) {
        this.next();
    }
}

prev() {
    this._slide(ORDER_PREV);
}

pause() {
    if (this._isSliding) {
        triggerTransitionEnd(this._element);
    }

    this._clearInterval();
}

cycle() {
    this._clearInterval();
}

```



```

    this._updateInterval();

    this._interval = setInterval(() => this.nextWhenVisible(), this._config.interval);
}

_maybeEnableCycle() {
    if (!this._config.ride) {
        return;
    }

    if (this._isSliding) {
        EventHandler.one(this._element, EVENT_SLID, () => this.cycle());
        return;
    }

    this.cycle();
}

to(index) {
    const items = this._getItems();

    if (index > items.length - 1 || index < 0) {
        return;
    }

    if (this._isSliding) {
        EventHandler.one(this._element, EVENT_SLID, () => this.to(index));
        return;
    }

    const activeIndex = this._getItemIndex(this._getActive());

    if (activeIndex === index) {
        return;
    }

    const order = index > activeIndex ? ORDER_NEXT : ORDER_PREV;

    this._slide(order, items[index]);
}

dispose() {
    if (this._swipeHelper) {
        this._swipeHelper.dispose();
    }

    super.dispose();
} // Private

_configAfterMerge(config) {
    config.defaultInterval = config.interval;
    return config;
}

_addEventListeners() {

```

```

    if (this._config.keyboard) {
      EventHandler.on(this._element, EVENT_KEYDOWN$1, event => this._keydown(event));
    }

    if (this._config.pause === 'hover') {
      EventHandler.on(this._element, EVENT_MOUSEENTER$1, () => this.pause());
      EventHandler.on(this._element, EVENT_MOUSELEAVE$1, () => this._maybeEnableCycle());
    }

    if (this._config.touch && Swipe.isSupported()) {
      this._addTouchEventListeners();
    }
  }

  _addTouchEventListeners() {
    for (const img of SelectorEngine.find(SELECTOR_ITEM_IMG, this._element)) {
      EventHandler.on(img, EVENT_DRAG_START, event => event.preventDefault());
    }

    const endCallBack = () => {
      if (this._config.pause !== 'hover') {
        return;
      }
      // If it's a touch-enabled device, mouseenter/leave are fired as
      // part of the mouse compatibility events on first tap - the carousel
      // would stop cycling until user tapped out of it;
      // here, we listen for touchend, explicitly pause the carousel
      // (as if it's the second time we tap on it, mouseenter compat event
      // is NOT fired) and after a timeout (to allow for mouse compatibility
      // events to fire) we explicitly restart cycling

      this.pause();

      if (this.touchTimeout) {
        clearTimeout(this.touchTimeout);
      }

      this.touchTimeout = setTimeout(() => this._maybeEnableCycle(), TOUCHEVENT_COMPAT_WAIT +
this._config.interval);
    };

    const swipeConfig = {
      leftCallback: () => this._slide(this._directionToOrder(DIRECTION_LEFT)),
      rightCallback: () => this._slide(this._directionToOrder(DIRECTION_RIGHT)),
      endCallback: endCallBack
    };
    this._swipeHelper = new Swipe(this._element, swipeConfig);
  }

  _keydown(event) {
    if (/input|textarea/i.test(event.target.tagName)) {
      return;
    }

    const direction = KEY_TO_DIRECTION[event.key];

    if (direction) {

```

```

    event.preventDefault();

    this._slide(this._directionToOrder(direction));
  }
}

_getItemIndex(element) {
  return this._getItems().indexOf(element);
}

_setActiveIndicatorElement(index) {
  if (!this._indicatorsElement) {
    return;
  }

  const activeIndicator = SelectorEngine.findOne(SELECTOR_ACTIVE, this._indicatorsElement);
  activeIndicator.classList.remove(CLASS_NAME_ACTIVE$2);
  activeIndicator.removeAttribute('aria-current');
  const newActiveIndicator = SelectorEngine.findOne(`[data-bs-slide-to="${index}"]`,
this._indicatorsElement);

  if (newActiveIndicator) {
    newActiveIndicator.classList.add(CLASS_NAME_ACTIVE$2);
    newActiveIndicator.setAttribute('aria-current', 'true');
  }
}

_updateInterval() {
  const element = this._activeElement || this._getActive();

  if (!element) {
    return;
  }

  const elementInterval = Number.parseInt(element.getAttribute('data-bs-interval'), 10);
  this._config.interval = elementInterval || this._config.defaultInterval;
}

_slide(order, element = null) {
  if (this._isSliding) {
    return;
  }

  const activeElement = this._getActive();

  const isNext = order === ORDER_NEXT;
  const nextElement = element || getNextActiveElement(this._getItems(), activeElement, isNext,
this._config.wrap);

  if (nextElement === activeElement) {
    return;
  }

  const nextElementIndex = this._getItemIndex(nextElement);

  const triggerEvent = eventName => {
    return EventHandler.trigger(this._element, eventName, {

```

```

        relatedTarget: nextElement,
        direction: this._orderToDirection(order),
        from: this._getItemIndex(activeElement),
        to: nextElementIndex
    });
};

const slideEvent = triggerEvent(EVENT_SLIDE);

if (slideEvent.defaultPrevented) {
    return;
}

if (!activeElement || !nextElement) {
    // Some weirdness is happening, so we bail
    // todo: change tests that use empty divs to avoid this check
    return;
}

const isCycling = Boolean(this._interval);
this.pause();
this._isSliding = true;

this._setActiveIndicatorElement(nextElementIndex);

this._activeElement = nextElement;
const directionalClassName = isNext ? CLASS_NAME_START : CLASS_NAME_END;
const orderClassName = isNext ? CLASS_NAME_NEXT : CLASS_NAME_PREV;
nextElement.classList.add(orderClassName);
reflow(nextElement);
activeElement.classList.add(directionalClassName);
nextElement.classList.add(directionalClassName);

const completeCallBack = () => {
    nextElement.classList.remove(directionalClassName, orderClassName);
    nextElement.classList.add(CLASS_NAME_ACTIVE$2);
    activeElement.classList.remove(CLASS_NAME_ACTIVE$2, orderClassName, directionalClassName);
    this._isSliding = false;
    triggerEvent(EVENT_SLID);
};

this._queueCallback(completeCallBack, activeElement, this._isAnimated());

if (isCycling) {
    this.cycle();
}
}

_isAnimated() {
    return this._element.classList.contains(CLASS_NAME_SLIDE);
}

_getActive() {
    return SelectorEngine.findOne(SELECTOR_ACTIVE_ITEM, this._element);
}

_getItems() {

```

```

    return SelectorEngine.find(SELECTOR_ITEM, this._element);
}

_clearInterval() {
  if (this._interval) {
    clearInterval(this._interval);
    this._interval = null;
  }
}

_directionToOrder(direction) {
  if (isRTL()) {
    return direction === DIRECTION_LEFT ? ORDER_PREV : ORDER_NEXT;
  }

  return direction === DIRECTION_LEFT ? ORDER_NEXT : ORDER_PREV;
}

_orderToDirection(order) {
  if (isRTL()) {
    return order === ORDER_PREV ? DIRECTION_LEFT : DIRECTION_RIGHT;
  }

  return order === ORDER_PREV ? DIRECTION_RIGHT : DIRECTION_LEFT;
} // Static

static jQueryInterface(config) {
  return this.each(function () {
    const data = Carousel.getInstance(this, config);

    if (typeof config === 'number') {
      data.to(config);
      return;
    }

    if (typeof config === 'string') {
      if (data[config] === undefined || config.startsWith('_') || config === 'constructor') {
        throw new TypeError(`No method named "${config}"`);
      }

      data[config]();
    }
  });
}

/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API$5, SELECTOR_DATA_SLIDE, function (event) {
  const target = getElementFromSelector(this);

  if (!target || !target.classList.contains(CLASS_NAME_CAROUSEL)) {
    return;
  }

```

```

    }

    event.preventDefault();
    const carousel = Carousel.getOrCreateInstance(target);
    const slideIndex = this.getAttribute('data-bs-slide-to');

    if (slideIndex) {
        carousel.to(slideIndex);

        carousel._maybeEnableCycle();

        return;
    }

    if (Manipulator.getDataAttribute(this, 'slide') === 'next') {
        carousel.next();

        carousel._maybeEnableCycle();

        return;
    }

    carousel.prev();

    carousel._maybeEnableCycle();
});
EventHandler.on(window, EVENT_LOAD_DATA_API$3, () => {
    const carousels = SelectorEngine.find(SELECTOR_DATA_RIDE);

    for (const carousel of carousels) {
        Carousel.getOrCreateInstance(carousel);
    }
});
/**
 * jQuery
 */

defineJQueryPlugin(Carousel);

/**
 * -----
 * Bootstrap (v5.2.2): collapse.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$b = 'collapse';
const DATA_KEY$7 = 'bs.collapse';
const EVENT_KEY$7 = `.${DATA_KEY$7}`;
const DATA_API_KEY$4 = '.data-api';
const EVENT_SHOW$6 = `show${EVENT_KEY$7}`;
const EVENT_SHOWN$6 = `shown${EVENT_KEY$7}`;
const EVENT_HIDE$6 = `hide${EVENT_KEY$7}`;
const EVENT_HIDDEN$6 = `hidden${EVENT_KEY$7}`;

```

```

const EVENT_CLICK_DATA_API$4 = `click${EVENT_KEY$7}${DATA_API_KEY$4}`;
const CLASS_NAME_SHOW$7 = 'show';
const CLASS_NAME_COLLAPSE = 'collapse';
const CLASS_NAME_COLLAPSING = 'collapsing';
const CLASS_NAME_COLLAPSED = 'collapsed';
const CLASS_NAME_DEEPER_CHILDREN =
`scope ${CLASS_NAME_COLLAPSE} ${CLASS_NAME_COLLAPSE}`;
const CLASS_NAME_HORIZONTAL = 'collapse-horizontal';
const WIDTH = 'width';
const HEIGHT = 'height';
const SELECTOR_ACTIVES = '.collapse.show, .collapse.collapsing';
const SELECTOR_DATA_TOGGLE$4 = '[data-bs-toggle="collapse"]';
const Default$a = {
  parent: null,
  toggle: true
};
const DefaultType$a = {
  parent: '(null|element)',
  toggle: 'boolean'
};
/**
 * Class definition
 */

class Collapse extends BaseComponent {
  constructor(element, config) {
    super(element, config);
    this._isTransitioning = false;
    this._triggerArray = [];
    const toggleList = SelectorEngine.find(SELECTOR_DATA_TOGGLE$4);

    for (const elem of toggleList) {
      const selector = getSelectorFromElement(elem);
      const filterElement = SelectorEngine.find(selector).filter(foundElement => foundElement ===
this._element);

      if (selector !== null && filterElement.length) {
        this._triggerArray.push(elem);
      }
    }

    this._initializeChildren();

    if (!this._config.parent) {
      this._addAriaAndCollapsedClass(this._triggerArray, this._isShown());
    }

    if (this._config.toggle) {
      this.toggle();
    }
  } // Getters

  static get Default() {
    return Default$a;
  }
}

```

```

static get DefaultType() {
  return DefaultType$a;
}

static get NAME() {
  return NAME$b;
} // Public

toggle() {
  if (this._isShown()) {
    this.hide();
  } else {
    this.show();
  }
}

show() {
  if (this._isTransitioning || this._isShown()) {
    return;
  }

  let activeChildren = []; // find active children

  if (this._config.parent) {
    activeChildren = this._getFirstLevelChildren(SELECTOR_ACTIVES).filter(element => element !==
this._element).map(element => Collapse.getOrCreateInstance(element, {
  toggle: false
}));
  }

  if (activeChildren.length && activeChildren[0]._isTransitioning) {
    return;
  }

  const startEvent = EventHandler.trigger(this._element, EVENT_SHOW$6);

  if (startEvent.defaultPrevented) {
    return;
  }

  for (const activeInstance of activeChildren) {
    activeInstance.hide();
  }

  const dimension = this._getDimension();

  this._element.classList.remove(CLASS_NAME_COLLAPSE);

  this._element.classList.add(CLASS_NAME_COLLAPSING);

  this._element.style[dimension] = 0;

  this._addAriaAndCollapsedClass(this._triggerArray, true);

  this._isTransitioning = true;

```



```

const complete = () => {
  this._isTransitioning = false;

  this._element.classList.remove(CLASS_NAME_COLLAPSING);

  this._element.classList.add(CLASS_NAME_COLLAPSE, CLASS_NAME_SHOW$7);

  this._element.style[dimension] = "";
  EventHandler.trigger(this._element, EVENT_SHOWN$6);
};

const capitalizedDimension = dimension[0].toUpperCase() + dimension.slice(1);
const scrollSize = `scroll${capitalizedDimension}`;

this._queueCallback(complete, this._element, true);

this._element.style[dimension] = `${this._element[scrollSize]}px`;
}

hide() {
  if (this._isTransitioning || !this._isShown()) {
    return;
  }

  const startEvent = EventHandler.trigger(this._element, EVENT_HIDE$6);

  if (startEvent.defaultPrevented) {
    return;
  }

  const dimension = this._getDimension();

  this._element.style[dimension] = `${this._element.getBoundingClientRect()[dimension]}px`;
  reflow(this._element);

  this._element.classList.add(CLASS_NAME_COLLAPSING);

  this._element.classList.remove(CLASS_NAME_COLLAPSE, CLASS_NAME_SHOW$7);

  for (const trigger of this._triggerArray) {
    const element = getElementFromSelector(trigger);

    if (element && !this._isShown(element)) {
      this._addAriaAndCollapsedClass([trigger], false);
    }
  }

  this._isTransitioning = true;

  const complete = () => {
    this._isTransitioning = false;

    this._element.classList.remove(CLASS_NAME_COLLAPSING);

    this._element.classList.add(CLASS_NAME_COLLAPSE);

    EventHandler.trigger(this._element, EVENT_HIDDEN$6);
  };

```

```

};

this._element.style[dimension] = "";

this._queueCallback(complete, this._element, true);
}

_isShown(element = this._element) {
  return element.classList.contains(CLASS_NAME_SHOW$7);
} // Private

_configAfterMerge(config) {
  config.toggle = Boolean(config.toggle); // Coerce string values

  config.parent = getElement(config.parent);
  return config;
}

_getDimension() {
  return this._element.classList.contains(CLASS_NAME_HORIZONTAL) ? WIDTH : HEIGHT;
}

_initializeChildren() {
  if (!this._config.parent) {
    return;
  }
  const children = this._getFirstLevelChildren(SELECTOR_DATA_TOGGLE$4);

  for (const element of children) {
    const selected = getElementFromSelector(element);

    if (selected) {
      this._addAriaAndCollapsedClass([element], this._isShown(selected));
    }
  }
}

_getFirstLevelChildren(selector) {
  const children = SelectorEngine.find(CLASS_NAME_DEEPER_CHILDREN, this._config.parent); //
  remove children if greater depth

  return SelectorEngine.find(selector, this._config.parent).filter(element
=> !children.includes(element));
}

_addAriaAndCollapsedClass(triggerArray, isOpen) {
  if (!triggerArray.length) {
    return;
  }

  for (const element of triggerArray) {
    element.classList.toggle(CLASS_NAME_COLLAPSED, !isOpen);
    element.setAttribute('aria-expanded', isOpen);
  }
} // Static

```

```

static jQueryInterface(config) {
  const _config = {};

  if (typeof config === 'string' && /show|hide/.test(config)) {
    _config.toggle = false;
  }

  return this.each(function () {
    const data = Collapse.getOrCreateInstance(this, _config);

    if (typeof config === 'string') {
      if (typeof data[config] === 'undefined') {
        throw new TypeError(`No method named "${config}"`);
      }

      data[config]();
    }
  });
}

/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API$4, SELECTOR_DATA_TOGGLE$4, function
(event) {
  // preventDefault only for <a> elements (which change the URL) not inside the collapsible element
  if (event.target.tagName === 'A' || event.delegateTarget && event.delegateTarget.tagName === 'A')
  {
    event.preventDefault();
  }

  const selector = getSelectorFromElement(this);
  const selectorElements = SelectorEngine.find(selector);

  for (const element of selectorElements) {
    Collapse.getOrCreateInstance(element, {
      toggle: false
    }).toggle();
  }
});
/**
 * jQuery
 */

definejQueryPlugin(Collapse);

var top = 'top';
var bottom = 'bottom';
var right = 'right';
var left = 'left';
var auto = 'auto';
var basePlacements = [top, bottom, right, left];

```

```

var start = 'start';
var end = 'end';
var clippingParents = 'clippingParents';
var viewport = 'viewport';
var popper = 'popper';
var reference = 'reference';
var variationPlacements = /*#__PURE__*/basePlacements.reduce(function (acc, placement) {
  return acc.concat([placement + "-" + start, placement + "-" + end]);
}, []);
var placements = /*#__PURE__*/[].concat(basePlacements, [auto]).reduce(function (acc, placement)
{
  return acc.concat([placement, placement + "-" + start, placement + "-" + end]);
}, []); // modifiers that need to read the DOM

var beforeRead = 'beforeRead';
var read = 'read';
var afterRead = 'afterRead'; // pure-logic modifiers

var beforeMain = 'beforeMain';
var main = 'main';
var afterMain = 'afterMain'; // modifier with the purpose to write to the DOM (or write into a
framework state)

var beforeWrite = 'beforeWrite';
var write = 'write';
var afterWrite = 'afterWrite';
var modifierPhases = [beforeRead, read, afterRead, beforeMain, main, afterMain, beforeWrite, write,
afterWrite];

function getNodeName(element) {
  return element ? (element.nodeName || "").toLowerCase() : null;
}

function getWindow(node) {
  if (node == null) {
    return window;
  }

  if (node.toString() !== '[object Window]') {
    var ownerDocument = node.ownerDocument;
    return ownerDocument ? ownerDocument.defaultView || window : window;
  }

  return node;
}

function isElement(node) {
  var OwnElement = getWindow(node).Element;
  return node instanceof OwnElement || node instanceof Element;
}

function isHTMLElement(node) {
  var OwnElement = getWindow(node).HTMLElement;
  return node instanceof OwnElement || node instanceof HTMLElement;
}

function isShadowRoot(node) {

```

```

// IE 11 has no ShadowRoot
if (typeof ShadowRoot === 'undefined') {
  return false;
}

var OwnElement = getWindow(node).ShadowRoot;
return node instanceof OwnElement || node instanceof ShadowRoot;
}

// and applies them to the HTMLElements such as popper and arrow

function applyStyles(_ref) {
  var state = _ref.state;
  Object.keys(state.elements).forEach(function (name) {
    var style = state.styles[name] || {};
    var attributes = state.attributes[name] || {};
    var element = state.elements[name]; // arrow is optional + virtual elements

    if (!isHTMLElement(element) || !getNodeName(element)) {
      return;
    }
    // Flow doesn't support to extend this property, but it's the most
    // effective way to apply styles to an HTMLElement
    // $FlowFixMe[cannot-write]

    Object.assign(element.style, style);
    Object.keys(attributes).forEach(function (name) {
      var value = attributes[name];

      if (value === false) {
        element.removeAttribute(name);
      } else {
        element.setAttribute(name, value === true ? '' : value);
      }
    });
  });
}

function effect$2(_ref2) {
  var state = _ref2.state;
  var initialStyles = {
    popper: {
      position: state.options.strategy,
      left: '0',
      top: '0',
      margin: '0'
    },
    arrow: {
      position: 'absolute'
    },
    reference: {}
  };
  Object.assign(state.elements.popper.style, initialStyles.popper);
  state.styles = initialStyles;

  if (state.elements.arrow) {
    Object.assign(state.elements.arrow.style, initialStyles.arrow);
  }
}

```

```

    }

    return function () {
      Object.keys(state.elements).forEach(function (name) {
        var element = state.elements[name];
        var attributes = state.attributes[name] || {};
        var styleProperties = Object.keys(state.styles.hasOwnProperty(name) ? state.styles[name] :
initialStyles[name]); // Set all values to an empty string to unset them

        var style = styleProperties.reduce(function (style, property) {
          style[property] = "";
          return style;
        }, {}); // arrow is optional + virtual elements

        if (!isHTMLElement(element) || !getNodeName(element)) {
          return;
        }

        Object.assign(element.style, style);
        Object.keys(attributes).forEach(function (attribute) {
          element.removeAttribute(attribute);
        });
      });
    };
  } // eslint-disable-next-line import/no-unused-modules

```

```

const applyStyles$1 = {
  name: 'applyStyles',
  enabled: true,
  phase: 'write',
  fn: applyStyles,
  effect: effect$2,
  requires: ['computeStyles']
};

```

```

function getBasePlacement(placement) {
  return placement.split('-')[0];
}

```

```

var max = Math.max;
var min = Math.min;
var round = Math.round;

```

```

function getUAString() {
  var uaData = navigator.userAgentData;

  if (uaData != null && uaData.brands) {
    return uaData.brands.map(function (item) {
      return item.brand + "/" + item.version;
    }).join(' ');
  }

```

```

  return navigator.userAgent;
}

```

```

function isLayoutViewport() {

```

```

    return !/^(?!chrome|android).*safari/i.test(getUAString());
}

function getBoundingClientRect(element, includeScale, isFixedStrategy) {
  if (includeScale === void 0) {
    includeScale = false;
  }

  if (isFixedStrategy === void 0) {
    isFixedStrategy = false;
  }

  var clientRect = element.getBoundingClientRect();
  var scaleX = 1;
  var scaleY = 1;

  if (includeScale && isHTMLElement(element)) {
    scaleX = element.offsetWidth > 0 ? round(clientRect.width) / element.offsetWidth || 1 : 1;
    scaleY = element.offsetHeight > 0 ? round(clientRect.height) / element.offsetHeight || 1 : 1;
  }

  var _ref = isElement(element) ? getWindow(element) : window,
      visualViewport = _ref.visualViewport;

  var addVisualOffsets = !isLayoutViewport() && isFixedStrategy;
  var x = (clientRect.left + (addVisualOffsets && visualViewport ? visualViewport.offsetLeft : 0)) /
scaleX;
  var y = (clientRect.top + (addVisualOffsets && visualViewport ? visualViewport.offsetTop : 0)) /
scaleY;
  var width = clientRect.width / scaleX;
  var height = clientRect.height / scaleY;
  return {
    width: width,
    height: height,
    top: y,
    right: x + width,
    bottom: y + height,
    left: x,
    x: x,
    y: y
  };
}

// means it doesn't take into account transforms.

function getLayoutRect(element) {
  var clientRect = getBoundingClientRect(element); // Use the clientRect sizes if it's not been
transformed.
  // Fixes https://github.com/popperjs/popper-core/issues/1223

  var width = element.offsetWidth;
  var height = element.offsetHeight;

  if (Math.abs(clientRect.width - width) <= 1) {
    width = clientRect.width;
  }
}

```

```

if (Math.abs(clientRect.height - height) <= 1) {
  height = clientRect.height;
}

return {
  x: element.offsetLeft,
  y: element.offsetTop,
  width: width,
  height: height
};
}

function contains(parent, child) {
  var rootNode = child.getRootNode && child.getRootNode(); // First, attempt with faster native
method

  if (parent.contains(child)) {
    return true;
  } // then fallback to custom implementation with Shadow DOM support
  else if (rootNode && isShadowRoot(rootNode)) {
    var next = child;

    do {
      if (next && parent.isSameNode(next)) {
        return true;
      } // $FlowFixMe[prop-missing]: need a better way to handle this...

      next = next.parentNode || next.host;
    } while (next);
  } // Give up, the result is false

  return false;
}

function getComputedStyle$1(element) {
  return getWindow(element).getComputedStyle(element);
}

function isTableElement(element) {
  return ['table', 'td', 'th'].indexOf(getNodeName(element)) >= 0;
}

function getDocumentElement(element) {
  // $FlowFixMe[incompatible-return]: assume body is always available
  return ((isElement(element) ? element.ownerDocument : // $FlowFixMe[prop-missing]
element.document) || window.document).documentElement;
}

function getParentNode(element) {
  if (getNodeName(element) === 'html') {
    return element;
  }

  return (// this is a quicker (but less type safe) way to save quite some bytes from the bundle
    // $FlowFixMe[incompatible-return]

```



```

    // $FlowFixMe[prop-missing]
    element.assignedSlot || // step into the shadow DOM of the parent of a slotted node
    element.parentNode || ( // DOM Element detected
    isShadowRoot(element) ? element.host : null) || // ShadowRoot detected
    // $FlowFixMe[incompatible-call]: HTMLElement is a Node
    getDocumentElement(element) // fallback

  );
}

function getTrueOffsetParent(element) {
  if (!isHTMLElement(element) || // https://github.com/popperjs/popper-core/issues/837
  getComputedStyle$1(element).position === 'fixed') {
    return null;
  }

  return element.offsetParent;
} // `.offsetParent` reports `null` for fixed elements, while absolute elements
// return the containing block

function getContainingBlock(element) {
  var isFirefox = /firefox/i.test(getUAString());
  var isIE = /Trident/i.test(getUAString());

  if (isIE && isHTMLElement(element)) {
    // In IE 9, 10 and 11 fixed elements containing block is always established by the viewport
    var elementCss = getComputedStyle$1(element);

    if (elementCss.position === 'fixed') {
      return null;
    }
  }

  var currentNode = getParentNode(element);

  if (isShadowRoot(currentNode)) {
    currentNode = currentNode.host;
  }

  while (isHTMLElement(currentNode) && ['html', 'body'].indexOf(getNodeName(currentNode)) < 0)
  {
    var css = getComputedStyle$1(currentNode); // This is non-exhaustive but covers the most
    common CSS properties that
    // create a containing block.
    // https://developer.mozilla.org/en-US/docs/Web/CSS/Containing_block#identifying_the_containing_block

    if (css.transform !== 'none' || css.perspective !== 'none' || css.contain === 'paint' || ['transform',
    'perspective'].indexOf(css.willChange) !== -1 || isFirefox && css.willChange === 'filter' || isFirefox &&
    css.filter && css.filter !== 'none') {
      return currentNode;
    } else {
      currentNode = currentNode.parentNode;
    }
  }
}

```

```

    return null;
} // Gets the closest ancestor positioned element. Handles some edge cases,
// such as table ancestors and cross browser bugs.

function getOffsetParent(element) {
    var window = getWindow(element);
    var offsetParent = getTrueOffsetParent(element);

    while (offsetParent && isTableElement(offsetParent) &&
getComputedStyle$1(offsetParent).position === 'static') {
        offsetParent = getTrueOffsetParent(offsetParent);
    }

    if (offsetParent && (getNodeName(offsetParent) === 'html' || getNodeName(offsetParent) ===
'body' && getComputedStyle$1(offsetParent).position === 'static')) {
        return window;
    }

    return offsetParent || getContainingBlock(element) || window;
}

function getMainAxisFromPlacement(placement) {
    return ['top', 'bottom'].indexOf(placement) >= 0 ? 'x' : 'y';
}

function within(min$1, value, max$1) {
    return max(min$1, min(value, max$1));
}
function withinMaxClamp(min, value, max) {
    var v = within(min, value, max);
    return v > max ? max : v;
}

function getFreshSideObject() {
    return {
        top: 0,
        right: 0,
        bottom: 0,
        left: 0
    };
}

function mergePaddingObject(paddingObject) {
    return Object.assign({}, getFreshSideObject(), paddingObject);
}

function expandToHashMap(value, keys) {
    return keys.reduce(function (hashMap, key) {
        hashMap[key] = value;
        return hashMap;
    }, {});
}

var toPaddingObject = function toPaddingObject(padding, state) {
    padding = typeof padding === 'function' ? padding(Object.assign({}, state.rects, {
        placement: state.placement

```

```

    ))) : padding;
    return mergePaddingObject(typeof padding !== 'number' ? padding : expandToHashMap(padding,
basePlacements));
  };

function arrow(_ref) {
  var _state$modifiersData$;

  var state = _ref.state,
      name = _ref.name,
      options = _ref.options;
  var arrowElement = state.elements.arrow;
  var popperOffsets = state.modifiersData.popperOffsets;
  var basePlacement = getBasePlacement(state.placement);
  var axis = getMainAxisFromPlacement(basePlacement);
  var isVertical = [left, right].indexOf(basePlacement) >= 0;
  var len = isVertical ? 'height' : 'width';

  if (!arrowElement || !popperOffsets) {
    return;
  }

  var paddingObject = toPaddingObject(options.padding, state);
  var arrowRect = getLayoutRect(arrowElement);
  var minProp = axis === 'y' ? top : left;
  var maxProp = axis === 'y' ? bottom : right;
  var endDiff = state.rects.reference[len] + state.rects.reference[axis] - popperOffsets[axis] -
state.rects.popper[len];
  var startDiff = popperOffsets[axis] - state.rects.reference[axis];
  var arrowOffsetParent = getOffsetParent(arrowElement);
  var clientSize = arrowOffsetParent ? axis === 'y' ? arrowOffsetParent.clientHeight || 0 :
arrowOffsetParent.clientWidth || 0 : 0;
  var centerToReference = endDiff / 2 - startDiff / 2; // Make sure the arrow doesn't overflow the
popper if the center point is
  // outside of the popper bounds

  var min = paddingObject[minProp];
  var max = clientSize - arrowRect[len] - paddingObject[maxProp];
  var center = clientSize / 2 - arrowRect[len] / 2 + centerToReference;
  var offset = within(min, center, max); // Prevents breaking syntax highlighting...

  var axisProp = axis;
  state.modifiersData[name] = (_state$modifiersData$ = {}, _state$modifiersData$[axisProp] = offset,
_state$modifiersData$.centerOffset = offset - center, _state$modifiersData$);
}

function effect$1(_ref2) {
  var state = _ref2.state,
      options = _ref2.options;
  var _options$element = options.element,
      arrowElement = _options$element === void 0 ? '[data-popper-arrow]' : _options$element;

  if (arrowElement == null) {
    return;
  } // CSS selector

```

```

if (typeof arrowElement === 'string') {
  arrowElement = state.elements.popper.querySelector(arrowElement);

  if (!arrowElement) {
    return;
  }
}

if (!contains(state.elements.popper, arrowElement)) {

  return;
}

state.elements.arrow = arrowElement;
} // eslint-disable-next-line import/no-unused-modules

```

```

const arrow$1 = {
  name: 'arrow',
  enabled: true,
  phase: 'main',
  fn: arrow,
  effect: effect$1,
  requires: ['popperOffsets'],
  requiresIfExists: ['preventOverflow']
};

```

```

function getVariation(placement) {
  return placement.split('-')[1];
}

```

```

var unsetSides = {
  top: 'auto',
  right: 'auto',
  bottom: 'auto',
  left: 'auto'
}; // Round the offsets to the nearest suitable subpixel based on the DPR.
// Zooming can change the DPR, but it seems to report a value that will
// cleanly divide the values into the appropriate subpixels.

```

```

function roundOffsetsByDPR(_ref) {
  var x = _ref.x,
      y = _ref.y;
  var win = window;
  var dpr = win.devicePixelRatio || 1;
  return {
    x: round(x * dpr) / dpr || 0,
    y: round(y * dpr) / dpr || 0
  };
}

```

```

function mapToStyles(_ref2) {
  var _Object$assign2;

  var popper = _ref2.popper,
      popperRect = _ref2.popperRect,
      placement = _ref2.placement,

```

```

    variation = _ref2.variation,
    offsets = _ref2.offsets,
    position = _ref2.position,
    gpuAcceleration = _ref2.gpuAcceleration,
    adaptive = _ref2.adaptive,
    roundOffsets = _ref2.roundOffsets,
    isFixed = _ref2.isFixed;
var _offsets$x = offsets.x,
    x = _offsets$x === void 0 ? 0 : _offsets$x,
    _offsets$y = offsets.y,
    y = _offsets$y === void 0 ? 0 : _offsets$y;

var _ref3 = typeof roundOffsets === 'function' ? roundOffsets({
  x: x,
  y: y
}): {
  x: x,
  y: y
};

x = _ref3.x;
y = _ref3.y;
var hasX = offsets.hasOwnProperty('x');
var hasY = offsets.hasOwnProperty('y');
var sideX = left;
var sideY = top;
var win = window;

if (adaptive) {
  var offsetParent = getOffsetParent/popper);
  var heightProp = 'clientHeight';
  var widthProp = 'clientWidth';

  if (offsetParent === getWindow/popper)) {
    offsetParent = getDocumentElement/popper);

    if (getComputedStyle$1(offsetParent).position !== 'static' && position === 'absolute') {
      heightProp = 'scrollHeight';
      widthProp = 'scrollWidth';
    }
  } // $FlowFixMe[incompatible-cast]: force type refinement, we compare offsetParent with window
  above, but Flow doesn't detect it

  offsetParent = offsetParent;

  if (placement === top || (placement === left || placement === right) && variation === end) {
    sideY = bottom;
    var offsetY = isFixed && offsetParent === win && win.visualViewport ? win.visualViewport.height :
  // $FlowFixMe[prop-missing]
    offsetParent[heightProp];
    y -= offsetY - popperRect.height;
    y *= gpuAcceleration ? 1 : -1;
  }

  if (placement === left || (placement === top || placement === bottom) && variation === end) {
    sideX = right;

```

```

    var offsetX = isFixed && offsetParent === win && win.visualViewport ? win.visualViewport.width :
// $FlowFixMe[prop-missing]
    offsetParent[widthProp];
    x -= offsetX - popperRect.width;
    x *= gpuAcceleration ? 1 : -1;
  }
}

var commonStyles = Object.assign({
  position: position
}, adaptive && unsetSides);

var _ref4 = roundOffsets === true ? roundOffsetsByDPR({
  x: x,
  y: y
}): {
  x: x,
  y: y
};

x = _ref4.x;
y = _ref4.y;

if (gpuAcceleration) {
  var _Object$assign;

  return Object.assign({}, commonStyles, (_Object$assign = {}, _Object$assign[sideY] = hasY ? '0' : '',
_Object$assign[sideX] = hasX ? '0' : '', _Object$assign.transform = (win.devicePixelRatio || 1) <= 1 ?
"translate(" + x + "px, " + y + "px)" : "translate3d(" + x + "px, " + y + "px, 0)", _Object$assign));
}

return Object.assign({}, commonStyles, (_Object$assign2 = {}, _Object$assign2[sideY] = hasY ? y +
"px" : "", _Object$assign2[sideX] = hasX ? x + "px" : "", _Object$assign2.transform = "",
_Object$assign2));
}

function computeStyles(_ref5) {
  var state = _ref5.state,
      options = _ref5.options;
  var _options$gpuAccelerat = options.gpuAcceleration,
      gpuAcceleration = _options$gpuAccelerat === void 0 ? true : _options$gpuAccelerat,
      _options$adaptive = options.adaptive,
      adaptive = _options$adaptive === void 0 ? true : _options$adaptive,
      _options$roundOffsets = options.roundOffsets,
      roundOffsets = _options$roundOffsets === void 0 ? true : _options$roundOffsets;

  var commonStyles = {
    placement: getBasePlacement(state.placement),
    variation: getVariation(state.placement),
    popper: state.elements.popper,
    popperRect: state.rects.popper,
    gpuAcceleration: gpuAcceleration,
    isFixed: state.options.strategy === 'fixed'
  };

  if (state.modifiersData.popperOffsets != null) {

```

```

    state.styles.popper = Object.assign({}, state.styles.popper, mapToStyles(Object.assign({},
commonStyles, {
    offsets: state.modifiersData.popperOffsets,
    position: state.options.strategy,
    adaptive: adaptive,
    roundOffsets: roundOffsets
    })));
}

if (state.modifiersData.arrow != null) {
    state.styles.arrow = Object.assign({}, state.styles.arrow, mapToStyles(Object.assign({},
commonStyles, {
    offsets: state.modifiersData.arrow,
    position: 'absolute',
    adaptive: false,
    roundOffsets: roundOffsets
    })));
}

state.attributes.popper = Object.assign({}, state.attributes.popper, {
    'data-popper-placement': state.placement
});
} // eslint-disable-next-line import/no-unused-modules

```

```

const computeStyles$1 = {
    name: 'computeStyles',
    enabled: true,
    phase: 'beforeWrite',
    fn: computeStyles,
    data: {}
};

```

```

var passive = {
    passive: true
};

```

```

function effect(_ref) {
    var state = _ref.state,
        instance = _ref.instance,
        options = _ref.options;
    var _options$scroll = options.scroll,
        scroll = _options$scroll === void 0 ? true : _options$scroll,
        _options$resize = options.resize,
        resize = _options$resize === void 0 ? true : _options$resize;
    var window = getWindow(state.elements.popper);
    var scrollParents = [].concat(state.scrollParents.reference, state.scrollParents.popper);

```

```

    if (scroll) {
        scrollParents.forEach(function (scrollParent) {
            scrollParent.addEventListener('scroll', instance.update, passive);
        });
    }

```

```

    if (resize) {
        window.addEventListener('resize', instance.update, passive);
    }

```

```

return function () {
  if (scroll) {
    scrollParents.forEach(function (scrollParent) {
      scrollParent.removeEventListener('scroll', instance.update, passive);
    });
  }

  if (resize) {
    window.removeEventListener('resize', instance.update, passive);
  }
};
} // eslint-disable-next-line import/no-unused-modules

```

```

const eventListeners = {
  name: 'eventListeners',
  enabled: true,
  phase: 'write',
  fn: function fn() {},
  effect: effect,
  data: {}
};

```

```

var hash$1 = {
  left: 'right',
  right: 'left',
  bottom: 'top',
  top: 'bottom'
};
function getOppositePlacement(placement) {
  return placement.replace(/left|right|bottom|top/g, function (matched) {
    return hash$1[matched];
  });
}

```

```

var hash = {
  start: 'end',
  end: 'start'
};
function getOppositeVariationPlacement(placement) {
  return placement.replace(/start|end/g, function (matched) {
    return hash[matched];
  });
}

```

```

function getWindowScroll(node) {
  var win = getWindow(node);
  var scrollLeft = win.pageXOffset;
  var scrollTop = win.pageYOffset;
  return {
    scrollLeft: scrollLeft,
    scrollTop: scrollTop
  };
}

```

```

function getWindowScrollBarX(element) {

```



```

// If <html> has a CSS width greater than the viewport, then this will be
// incorrect for RTL.
// Popper 1 is broken in this case and never had a bug report so let's assume
// it's not an issue. I don't think anyone ever specifies width on <html>
// anyway.
// Browsers where the left scrollbar doesn't cause an issue report `0` for
// this (e.g. Edge 2019, IE11, Safari)
return getBoundingClientRect(getDocumentElement(element)).left +
getWindowScroll(element).scrollLeft;
}

function getViewportRect(element, strategy) {
  var win = getWindow(element);
  var html = getDocumentElement(element);
  var visualViewport = win.visualViewport;
  var width = html.clientWidth;
  var height = html.clientHeight;
  var x = 0;
  var y = 0;

  if (visualViewport) {
    width = visualViewport.width;
    height = visualViewport.height;
    var layoutViewport = isLayoutViewport();

    if (layoutViewport || !layoutViewport && strategy === 'fixed') {
      x = visualViewport.offsetLeft;
      y = visualViewport.offsetTop;
    }
  }

  return {
    width: width,
    height: height,
    x: x + getWindowScrollBarX(element),
    y: y
  };
}

// of the `<html>` and `<body>` rect bounds if horizontally scrollable

function getDocumentRect(element) {
  var _element$ownerDocumen;

  var html = getDocumentElement(element);
  var winScroll = getWindowScroll(element);
  var body = (_element$ownerDocumen = element.ownerDocument) == null ? void 0 :
_element$ownerDocumen.body;
  var width = max(html.scrollWidth, html.clientWidth, body ? body.scrollWidth : 0, body ?
body.clientWidth : 0);
  var height = max(html.scrollHeight, html.clientHeight, body ? body.scrollHeight : 0, body ?
body.clientHeight : 0);
  var x = -winScroll.scrollLeft + getWindowScrollBarX(element);
  var y = -winScroll.scrollTop;

  if (getComputedStyle$1(body || html).direction === 'rtl') {
    x += max(html.clientWidth, body ? body.clientWidth : 0) - width;
  }
}

```

```

    }

    return {
        width: width,
        height: height,
        x: x,
        y: y
    };
}

function isScrollParent(element) {
    // Firefox wants us to check `x` and `y` variations as well
    var _getComputedStyle = getComputedStyle$1(element),
        overflow = _getComputedStyle.overflow,
        overflowX = _getComputedStyle.overflowX,
        overflowY = _getComputedStyle.overflowY;

    return /auto|scroll|overlay|hidden/.test(overflow + overflowY + overflowX);
}

function getScrollParent(node) {
    if (['html', 'body', '#document'].indexOf(getNodeName(node)) >= 0) {
        // $FlowFixMe[incompatible-return]: assume body is always available
        return node.ownerDocument.body;
    }

    if (isHTMLElement(node) && isScrollParent(node)) {
        return node;
    }

    return getScrollParent(getParentNode(node));
}

/*
given a DOM element, return the list of all scroll parents, up the list of ancestors
until we get to the top window object. This list is what we attach scroll listeners
to, because if any of these parent elements scroll, we'll need to re-calculate the
reference element's position.
*/

function listScrollParents(element, list) {
    var _element$ownerDocumen;

    if (list === void 0) {
        list = [];
    }

    var scrollParent = getScrollParent(element);
    var isBody = scrollParent === ((_element$ownerDocumen = element.ownerDocument) == null ?
void 0 : _element$ownerDocumen.body);
    var win = getWindow(scrollParent);
    var target = isBody ? [win].concat(win.visualViewport || [], isScrollParent(scrollParent) ?
scrollParent : []) : scrollParent;
    var updatedList = list.concat(target);
    return isBody ? updatedList : // $FlowFixMe[incompatible-call]: isBody tells us target will be an
HTMLElement here
        updatedList.concat(listScrollParents(getParentNode(target)));
}

```

```

}

function rectToClientRect(rect) {
  return Object.assign({}, rect, {
    left: rect.x,
    top: rect.y,
    right: rect.x + rect.width,
    bottom: rect.y + rect.height
  });
}

function getInnerBoundingClientRect(element, strategy) {
  var rect = getBoundingClientRect(element, false, strategy === 'fixed');
  rect.top = rect.top + element.clientTop;
  rect.left = rect.left + element.clientLeft;
  rect.bottom = rect.top + element.clientHeight;
  rect.right = rect.left + element.clientWidth;
  rect.width = element.clientWidth;
  rect.height = element.clientHeight;
  rect.x = rect.left;
  rect.y = rect.top;
  return rect;
}

function getClientRectFromMixedType(element, clippingParent, strategy) {
  return clippingParent === viewport ? rectToClientRect(getViewportRect(element, strategy)) :
  isElement(clippingParent) ? getInnerBoundingClientRect(clippingParent, strategy) :
  rectToClientRect(getDocumentRect(getDocumentElement(element)));
} // A "clipping parent" is an overflowable container with the characteristic of
// clipping (or hiding) overflowing elements with a position different from
// `initial`

function getClippingParents(element) {
  var clippingParents = listScrollParents(getParentNode(element));
  var canEscapeClipping = ['absolute', 'fixed'].indexOf(getComputedStyle$1(element).position) >= 0;
  var clipperElement = canEscapeClipping && isHTMLElement(element) ? getOffsetParent(element) :
  element;

  if (!isElement(clipperElement)) {
    return [];
  } // $FlowFixMe[incompatible-return]: https://github.com/facebook/flow/issues/1414

  return clippingParents.filter(function (clippingParent) {
    return isElement(clippingParent) && contains(clippingParent, clipperElement) &&
    getNodeName(clippingParent) !== 'body';
  });
} // Gets the maximum area that the element is visible in due to any number of
// clipping parents

function getClippingRect(element, boundary, rootBoundary, strategy) {
  var mainClippingParents = boundary === 'clippingParents' ? getClippingParents(element) :
  [].concat(boundary);
  var clippingParents = [].concat(mainClippingParents, [rootBoundary]);
  var firstClippingParent = clippingParents[0];

```

```

var clippingRect = clippingParents.reduce(function (accRect, clippingParent) {
  var rect = getClientRectFromMixedType(element, clippingParent, strategy);
  accRect.top = max(rect.top, accRect.top);
  accRect.right = min(rect.right, accRect.right);
  accRect.bottom = min(rect.bottom, accRect.bottom);
  accRect.left = max(rect.left, accRect.left);
  return accRect;
}, getClientRectFromMixedType(element, firstClippingParent, strategy));
clippingRect.width = clippingRect.right - clippingRect.left;
clippingRect.height = clippingRect.bottom - clippingRect.top;
clippingRect.x = clippingRect.left;
clippingRect.y = clippingRect.top;
return clippingRect;
}

```

```

function computeOffsets(_ref) {
  var reference = _ref.reference,
      element = _ref.element,
      placement = _ref.placement;
  var basePlacement = placement ? getBasePlacement(placement) : null;
  var variation = placement ? getVariation(placement) : null;
  var commonX = reference.x + reference.width / 2 - element.width / 2;
  var commonY = reference.y + reference.height / 2 - element.height / 2;
  var offsets;

```

```

  switch (basePlacement) {
    case top:
      offsets = {
        x: commonX,
        y: reference.y - element.height
      };
      break;

```

```

    case bottom:
      offsets = {
        x: commonX,
        y: reference.y + reference.height
      };
      break;

```

```

    case right:
      offsets = {
        x: reference.x + reference.width,
        y: commonY
      };
      break;

```

```

    case left:
      offsets = {
        x: reference.x - element.width,
        y: commonY
      };
      break;

```

```

    default:
      offsets = {
        x: reference.x,

```

```

    y: reference.y
  };
}

var mainAxis = basePlacement ? getMainAxisFromPlacement(basePlacement) : null;

if (mainAxis != null) {
  var len = mainAxis === 'y' ? 'height' : 'width';

  switch (variation) {
    case start:
      offsets[mainAxis] = offsets[mainAxis] - (reference[len] / 2 - element[len] / 2);
      break;

    case end:
      offsets[mainAxis] = offsets[mainAxis] + (reference[len] / 2 - element[len] / 2);
      break;
  }
}

return offsets;
}

function detectOverflow(state, options) {
  if (options === void 0) {
    options = {};
  }

  var _options = options,
      _options$placement = _options.placement,
      placement = _options$placement === void 0 ? state.placement : _options$placement,
      _options$strategy = _options.strategy,
      strategy = _options$strategy === void 0 ? state.strategy : _options$strategy,
      _options$boundary = _options.boundary,
      boundary = _options$boundary === void 0 ? clippingParents : _options$boundary,
      _options$rootBoundary = _options.rootBoundary,
      rootBoundary = _options$rootBoundary === void 0 ? viewport : _options$rootBoundary,
      _options$elementConte = _options.elementContext,
      elementContext = _options$elementConte === void 0 ? popper : _options$elementConte,
      _options$altBoundary = _options.altBoundary,
      altBoundary = _options$altBoundary === void 0 ? false : _options$altBoundary,
      _options$padding = _options.padding,
      padding = _options$padding === void 0 ? 0 : _options$padding;
  var paddingObject = mergePaddingObject(typeof padding !== 'number' ? padding :
expandToHashMap(padding, basePlacements));
  var altContext = elementContext === popper ? reference : popper;
  var popperRect = state.rects.popper;
  var element = state.elements[altBoundary ? altContext : elementContext];
  var clippingClientRect = getClippingRect(isElement(element) ? element : element.contextElement
|| getDocumentElement(state.elements.popper), boundary, rootBoundary, strategy);
  var referenceClientRect = getBoundingClientRect(state.elements.reference);
  var popperOffsets = computeOffsets({
    reference: referenceClientRect,
    element: popperRect,
    strategy: 'absolute',
    placement: placement
  });
};

```

```

    var popperClientRect = rectToClientRect(Object.assign({}, popperRect, popperOffsets));
    var elementClientRect = elementContext === popper ? popperClientRect : referenceClientRect; //
    positive = overflowing the clipping rect
    // 0 or negative = within the clipping rect

    var overflowOffsets = {
      top: clippingClientRect.top - elementClientRect.top + paddingObject.top,
      bottom: elementClientRect.bottom - clippingClientRect.bottom + paddingObject.bottom,
      left: clippingClientRect.left - elementClientRect.left + paddingObject.left,
      right: elementClientRect.right - clippingClientRect.right + paddingObject.right
    };
    var offsetData = state.modifiersData.offset; // Offsets can be applied only to the popper element

    if (elementContext === popper && offsetData) {
      var offset = offsetData[placement];
      Object.keys(overflowOffsets).forEach(function (key) {
        var multiply = [right, bottom].indexOf(key) >= 0 ? 1 : -1;
        var axis = [top, bottom].indexOf(key) >= 0 ? 'y' : 'x';
        overflowOffsets[key] += offset[axis] * multiply;
      });
    }

    return overflowOffsets;
  }

  function computeAutoPlacement(state, options) {
    if (options === void 0) {
      options = {};
    }

    var _options = options,
        placement = _options.placement,
        boundary = _options.boundary,
        rootBoundary = _options.rootBoundary,
        padding = _options.padding,
        flipVariations = _options.flipVariations,
        _options$allowedAutoP = _options.allowedAutoPlacements,
        allowedAutoPlacements = _options$allowedAutoP === void 0 ? placements :
    _options$allowedAutoP;
    var variation = getVariation(placement);
    var placements$1 = variation ? flipVariations ? variationPlacements :
    variationPlacements.filter(function (placement) {
      return getVariation(placement) === variation;
    }) : basePlacements;
    var allowedPlacements = placements$1.filter(function (placement) {
      return allowedAutoPlacements.indexOf(placement) >= 0;
    });

    if (allowedPlacements.length === 0) {
      allowedPlacements = placements$1;
    } // $FlowFixMe[incompatible-type]: Flow seems to have problems with two array unions...

    var overflows = allowedPlacements.reduce(function (acc, placement) {
      acc[placement] = detectOverflow(state, {
        placement: placement,
        boundary: boundary,

```

```

    rootBoundary: rootBoundary,
    padding: padding
  }[getBasePlacement(placement)]);
  return acc;
}, {});
return Object.keys(overflows).sort(function (a, b) {
  return overflows[a] - overflows[b];
});
}

function getExpandedFallbackPlacements(placement) {
  if (getBasePlacement(placement) === auto) {
    return [];
  }

  var oppositePlacement = getOppositePlacement(placement);
  return [getOppositeVariationPlacement(placement), oppositePlacement,
  getOppositeVariationPlacement(oppositePlacement)];
}

function flip(_ref) {
  var state = _ref.state,
      options = _ref.options,
      name = _ref.name;

  if (state.modifiersData[name]._skip) {
    return;
  }

  var _options$mainAxis = options.mainAxis,
      checkMainAxis = _options$mainAxis === void 0 ? true : _options$mainAxis,
      _options$altAxis = options.altAxis,
      checkAltAxis = _options$altAxis === void 0 ? true : _options$altAxis,
      specifiedFallbackPlacements = options.fallbackPlacements,
      padding = options.padding,
      boundary = options.boundary,
      rootBoundary = options.rootBoundary,
      altBoundary = options.altBoundary,
      _options$flipVariatio = options.flipVariations,
      flipVariations = _options$flipVariatio === void 0 ? true : _options$flipVariatio,
      allowedAutoPlacements = options.allowedAutoPlacements;
  var preferredPlacement = state.options.placement;
  var basePlacement = getBasePlacement(preferredPlacement);
  var isBasePlacement = basePlacement === preferredPlacement;
  var fallbackPlacements = specifiedFallbackPlacements || (isBasePlacement || !flipVariations ?
  [getOppositePlacement(preferredPlacement)] :
  getExpandedFallbackPlacements(preferredPlacement));
  var placements = [preferredPlacement].concat(fallbackPlacements).reduce(function (acc,
  placement) {
    return acc.concat(getBasePlacement(placement) === auto ? computeAutoPlacement(state, {
      placement: placement,
      boundary: boundary,
      rootBoundary: rootBoundary,
      padding: padding,
      flipVariations: flipVariations,
      allowedAutoPlacements: allowedAutoPlacements
    }) : placement);
  }, []);
}

```

```

}, []);
var referenceRect = state.rects.reference;
var popperRect = state.rects.popper;
var checksMap = new Map();
var makeFallbackChecks = true;
var firstFittingPlacement = placements[0];

for (var i = 0; i < placements.length; i++) {
  var placement = placements[i];

  var _basePlacement = getBasePlacement(placement);

  var isStartVariation = getVariation(placement) === start;
  var isVertical = [top, bottom].indexOf(_basePlacement) >= 0;
  var len = isVertical ? 'width' : 'height';
  var overflow = detectOverflow(state, {
    placement: placement,
    boundary: boundary,
    rootBoundary: rootBoundary,
    altBoundary: altBoundary,
    padding: padding
  });

  var mainVariationSide = isVertical ? isStartVariation ? right : left : isStartVariation ? bottom : top;

  if (referenceRect[len] > popperRect[len]) {
    mainVariationSide = getOppositePlacement(mainVariationSide);
  }

  var altVariationSide = getOppositePlacement(mainVariationSide);
  var checks = [];

  if (checkMainAxis) {
    checks.push(overflow[_basePlacement] <= 0);
  }

  if (checkAltAxis) {
    checks.push(overflow[mainVariationSide] <= 0, overflow[altVariationSide] <= 0);
  }

  if (checks.every(function (check) {
    return check;
  })) {
    firstFittingPlacement = placement;
    makeFallbackChecks = false;
    break;
  }

  checksMap.set(placement, checks);
}

if (makeFallbackChecks) {
  // `2` may be desired in some cases – research later
  var numberOfChecks = flipVariations ? 3 : 1;

  var _loop = function _loop(_i) {
    var fittingPlacement = placements.find(function (placement) {
      var checks = checksMap.get(placement);

```



```

    if (checks) {
      return checks.slice(0, _i).every(function (check) {
        return check;
      });
    }
  });

  if (fittingPlacement) {
    firstFittingPlacement = fittingPlacement;
    return "break";
  }
};

for (var _i = numberOfChecks; _i > 0; _i--) {
  var _ret = _loop(_i);

  if (_ret === "break") break;
}

if (state.placement !== firstFittingPlacement) {
  state.modifiersData[name]._skip = true;
  state.placement = firstFittingPlacement;
  state.reset = true;
}
} // eslint-disable-next-line import/no-unused-modules

```

```

const flip$1 = {
  name: 'flip',
  enabled: true,
  phase: 'main',
  fn: flip,
  requiresIfExists: ['offset'],
  data: {
    _skip: false
  }
};

```

```

function getSideOffsets(overflow, rect, preventedOffsets) {
  if (preventedOffsets === void 0) {
    preventedOffsets = {
      x: 0,
      y: 0
    };
  }

  return {
    top: overflow.top - rect.height - preventedOffsets.y,
    right: overflow.right - rect.width + preventedOffsets.x,
    bottom: overflow.bottom - rect.height + preventedOffsets.y,
    left: overflow.left - rect.width - preventedOffsets.x
  };
}

```

```

function isAnySideFullyClipped(overflow) {

```

```

return [top, right, bottom, left].some(function (side) {
  return overflow[side] >= 0;
});
}

function hide(_ref) {
  var state = _ref.state,
      name = _ref.name;
  var referenceRect = state.rects.reference;
  var popperRect = state.rects.popper;
  var preventedOffsets = state.modifiersData.preventOverflow;
  var referenceOverflow = detectOverflow(state, {
    elementContext: 'reference'
  });
  var popperAltOverflow = detectOverflow(state, {
    altBoundary: true
  });
  var referenceClippingOffsets = getSideOffsets(referenceOverflow, referenceRect);
  var popperEscapeOffsets = getSideOffsets(popperAltOverflow, popperRect, preventedOffsets);
  var isReferenceHidden = isAnySideFullyClipped(referenceClippingOffsets);
  var hasPopperEscaped = isAnySideFullyClipped(popperEscapeOffsets);
  state.modifiersData[name] = {
    referenceClippingOffsets: referenceClippingOffsets,
    popperEscapeOffsets: popperEscapeOffsets,
    isReferenceHidden: isReferenceHidden,
    hasPopperEscaped: hasPopperEscaped
  };
  state.attributes.popper = Object.assign({}, state.attributes.popper, {
    'data-popper-reference-hidden': isReferenceHidden,
    'data-popper-escaped': hasPopperEscaped
  });
} // eslint-disable-next-line import/no-unused-modules

```

```

const hide$1 = {
  name: 'hide',
  enabled: true,
  phase: 'main',
  requiresIfExists: ['preventOverflow'],
  fn: hide
};

```

```

function distanceAndSkiddingToXY(placement, rects, offset) {
  var basePlacement = getBasePlacement(placement);
  var invertDistance = [left, top].indexOf(basePlacement) >= 0 ? -1 : 1;

  var _ref = typeof offset === 'function' ? offset(Object.assign({}, rects, {
    placement: placement
  })) : offset,
      skidding = _ref[0],
      distance = _ref[1];

  skidding = skidding || 0;
  distance = (distance || 0) * invertDistance;
  return [left, right].indexOf(basePlacement) >= 0 ? {
    x: distance,
    y: skidding
  } : {
    x: skidding,
    y: distance
  };
}

```

```

    } : {
      x: skidding,
      y: distance
    };
  }

function offset(_ref2) {
  var state = _ref2.state,
      options = _ref2.options,
      name = _ref2.name;
  var _options$offset = options.offset,
      offset = _options$offset === void 0 ? [0, 0] : _options$offset;
  var data = placements.reduce(function (acc, placement) {
    acc[placement] = distanceAndSkiddingToXY(placement, state.rects, offset);
    return acc;
  }, {});
  var _data$state$placement = data[state.placement],
      x = _data$state$placement.x,
      y = _data$state$placement.y;

  if (state.modifiersData.popperOffsets != null) {
    state.modifiersData.popperOffsets.x += x;
    state.modifiersData.popperOffsets.y += y;
  }

  state.modifiersData[name] = data;
} // eslint-disable-next-line import/no-unused-modules

```

```

const offset$1 = {
  name: 'offset',
  enabled: true,
  phase: 'main',
  requires: ['popperOffsets'],
  fn: offset
};

```

```

function popperOffsets(_ref) {
  var state = _ref.state,
      name = _ref.name;
  // Offsets are the actual position the popper needs to have to be
  // properly positioned near its reference element
  // This is the most basic placement, and will be adjusted by
  // the modifiers in the next step
  state.modifiersData[name] = computeOffsets({
    reference: state.rects.reference,
    element: state.rects.popper,
    strategy: 'absolute',
    placement: state.placement
  });
} // eslint-disable-next-line import/no-unused-modules

```

```

const popperOffsets$1 = {
  name: 'popperOffsets',
  enabled: true,
  phase: 'read',

```

```

    fn: popperOffsets,
    data: {}
  };

  function getAltAxis(axis) {
    return axis === 'x' ? 'y' : 'x';
  }

  function preventOverflow(_ref) {
    var state = _ref.state,
        options = _ref.options,
        name = _ref.name;
    var _options$mainAxis = options.mainAxis,
        checkMainAxis = _options$mainAxis === void 0 ? true : _options$mainAxis,
        _options$altAxis = options.altAxis,
        checkAltAxis = _options$altAxis === void 0 ? false : _options$altAxis,
        boundary = options.boundary,
        rootBoundary = options.rootBoundary,
        altBoundary = options.altBoundary,
        padding = options.padding,
        _options$tether = options.tether,
        tether = _options$tether === void 0 ? true : _options$tether,
        _options$tetherOffset = options.tetherOffset,
        tetherOffset = _options$tetherOffset === void 0 ? 0 : _options$tetherOffset;
    var overflow = detectOverflow(state, {
      boundary: boundary,
      rootBoundary: rootBoundary,
      padding: padding,
      altBoundary: altBoundary
    });
    var basePlacement = getBasePlacement(state.placement);
    var variation = getVariation(state.placement);
    var isBasePlacement = !variation;
    var mainAxis = getMainAxisFromPlacement(basePlacement);
    var altAxis = getAltAxis(mainAxis);
    var popperOffsets = state.modifiersData.popperOffsets;
    var referenceRect = state.rects.reference;
    var popperRect = state.rects.popper;
    var tetherOffsetValue = typeof tetherOffset === 'function' ? tetherOffset(Object.assign({}, state.rects, {
      placement: state.placement
    })) : tetherOffset;
    var normalizedTetherOffsetValue = typeof tetherOffsetValue === 'number' ? {
      mainAxis: tetherOffsetValue,
      altAxis: tetherOffsetValue
    } : Object.assign({
      mainAxis: 0,
      altAxis: 0
    }, tetherOffsetValue);
    var offsetModifierState = state.modifiersData.offset ? state.modifiersData.offset[state.placement] : null;
    var data = {
      x: 0,
      y: 0
    };

    if (!popperOffsets) {

```

```

    return;
}

if (checkMainAxis) {
    var _offsetModifierState$;

    var mainSide = mainAxis === 'y' ? top : left;
    var altSide = mainAxis === 'y' ? bottom : right;
    var len = mainAxis === 'y' ? 'height' : 'width';
    var offset = popperOffsets[mainAxis];
    var min$1 = offset + overflow[mainSide];
    var max$1 = offset - overflow[altSide];
    var additive = tether ? -popperRect[len] / 2 : 0;
    var minLen = variation === start ? referenceRect[len] : popperRect[len];
    var maxLen = variation === start ? -popperRect[len] : -referenceRect[len]; // We need to include
the arrow in the calculation so the arrow doesn't go
    // outside the reference bounds

    var arrowElement = state.elements.arrow;
    var arrowRect = tether && arrowElement ? getLayoutRect(arrowElement) : {
        width: 0,
        height: 0
    };
    var arrowPaddingObject = state.modifiersData['arrow#persistent'] ?
state.modifiersData['arrow#persistent'].padding : getFreshSideObject();
    var arrowPaddingMin = arrowPaddingObject[mainSide];
    var arrowPaddingMax = arrowPaddingObject[altSide]; // If the reference length is smaller than the
arrow length, we don't want
    // to include its full size in the calculation. If the reference is small
    // and near the edge of a boundary, the popper can overflow even if the
    // reference is not overflowing as well (e.g. virtual elements with no
    // width or height)

    var arrowLen = within(0, referenceRect[len], arrowRect[len]);
    var minOffset = isBasePlacement ? referenceRect[len] / 2 - additive - arrowLen - arrowPaddingMin
- normalizedTetherOffsetValue.mainAxis : minLen - arrowLen - arrowPaddingMin -
normalizedTetherOffsetValue.mainAxis;
    var maxOffset = isBasePlacement ? -referenceRect[len] / 2 + additive + arrowLen +
arrowPaddingMax + normalizedTetherOffsetValue.mainAxis : maxLen + arrowLen + arrowPaddingMax
+ normalizedTetherOffsetValue.mainAxis;
    var arrowOffsetParent = state.elements.arrow && getOffsetParent(state.elements.arrow);
    var clientOffset = arrowOffsetParent ? mainAxis === 'y' ? arrowOffsetParent.clientTop || 0 :
arrowOffsetParent.clientLeft || 0 : 0;
    var offsetModifierValue = (_offsetModifierState$ = offsetModifierState == null ? void 0 :
offsetModifierState[mainAxis]) != null ? _offsetModifierState$ : 0;
    var tetherMin = offset + minOffset - offsetModifierValue - clientOffset;
    var tetherMax = offset + maxOffset - offsetModifierValue;
    var preventedOffset = within(tether ? min(min$1, tetherMin) : min$1, offset, tether ? max(max$1,
tetherMax) : max$1);
    popperOffsets[mainAxis] = preventedOffset;
    data[mainAxis] = preventedOffset - offset;
}

if (checkAltAxis) {
    var _offsetModifierState$2;

    var _mainSide = mainAxis === 'x' ? top : left;

```

```

var _altSide = mainAxis === 'x' ? bottom : right;

var _offset = popperOffsets[altAxis];

var _len = altAxis === 'y' ? 'height' : 'width';

var _min = _offset + overflow[_mainSide];

var _max = _offset - overflow[_altSide];

var isOriginSide = [top, left].indexOf(basePlacement) !== -1;

var _offsetModifierValue = (_offsetModifierState$2 = offsetModifierState == null ? void 0 : offsetModifierState[altAxis]) !== null ? _offsetModifierState$2 : 0;

var _tetherMin = isOriginSide ? _min : _offset - referenceRect[_len] - popperRect[_len] - _offsetModifierValue + normalizedTetherOffsetValue.altAxis;

var _tetherMax = isOriginSide ? _offset + referenceRect[_len] + popperRect[_len] - _offsetModifierValue - normalizedTetherOffsetValue.altAxis : _max;

var _preventedOffset = tether && isOriginSide ? withinMaxClamp(_tetherMin, _offset, _tetherMax) : within(tether ? _tetherMin : _min, _offset, tether ? _tetherMax : _max);

popperOffsets[altAxis] = _preventedOffset;
data[altAxis] = _preventedOffset - _offset;
}

state.modifiersData[name] = data;
} // eslint-disable-next-line import/no-unused-modules

const preventOverflow$1 = {
  name: 'preventOverflow',
  enabled: true,
  phase: 'main',
  fn: preventOverflow,
  requiresIfExists: ['offset']
};

function getHTMLElementScroll(element) {
  return {
    scrollLeft: element.scrollLeft,
    scrollTop: element.scrollTop
  };
}

function getNodeScroll(node) {
  if (node === getWindow(node) || !isHTMLElement(node)) {
    return getWindowScroll(node);
  } else {
    return getHTMLElementScroll(node);
  }
}

function isElementScaled(element) {

```

```

var rect = element.getBoundingClientRect();
var scaleX = round(rect.width) / element.offsetWidth || 1;
var scaleY = round(rect.height) / element.offsetHeight || 1;
return scaleX !== 1 || scaleY !== 1;
} // Returns the composite rect of an element relative to its offsetParent.
// Composite means it takes into account transforms as well as layout.

```

```

function getCompositeRect(elementOrVirtualElement, offsetParent, isFixed) {
  if (isFixed === void 0) {
    isFixed = false;
  }

  var isOffsetParentAnElement = isHTMLElement(offsetParent);
  var offsetParentIsScaled = isHTMLElement(offsetParent) && isElementScaled(offsetParent);
  var documentElement = getDocumentElement(offsetParent);
  var rect = getBoundingClientRect(elementOrVirtualElement, offsetParentIsScaled, isFixed);
  var scroll = {
    scrollLeft: 0,
    scrollTop: 0
  };
  var offsets = {
    x: 0,
    y: 0
  };

  if (isOffsetParentAnElement || !isOffsetParentAnElement && !isFixed) {
    if (getNodeName(offsetParent) !== 'body' || // https://github.com/popperjs/popper-core/issues/1078
        isScrollParent(documentElement)) {
      scroll = getNodeScroll(offsetParent);
    }

    if (isHTMLElement(offsetParent)) {
      offsets = getBoundingClientRect(offsetParent, true);
      offsets.x += offsetParent.clientLeft;
      offsets.y += offsetParent.clientTop;
    } else if (documentElement) {
      offsets.x = getWindowScrollBarX(documentElement);
    }
  }

  return {
    x: rect.left + scroll.scrollLeft - offsets.x,
    y: rect.top + scroll.scrollTop - offsets.y,
    width: rect.width,
    height: rect.height
  };
}

function order(modifiers) {
  var map = new Map();
  var visited = new Set();
  var result = [];
  modifiers.forEach(function (modifier) {
    map.set(modifier.name, modifier);
  }); // On visiting object, check for its dependencies and visit them recursively

```

```

function sort(modifier) {
  visited.add(modifier.name);
  var requires = [].concat(modifier.requires || [], modifier.requiresIfExists || []);
  requires.forEach(function (dep) {
    if (!visited.has(dep)) {
      var depModifier = map.get(dep);

      if (depModifier) {
        sort(depModifier);
      }
    }
  });
  result.push(modifier);
}

modifiers.forEach(function (modifier) {
  if (!visited.has(modifier.name)) {
    // check for visited object
    sort(modifier);
  }
});
return result;
}

function orderModifiers(modifiers) {
  // order based on dependencies
  var orderedModifiers = order(modifiers); // order based on phase

  return modifierPhases.reduce(function (acc, phase) {
    return acc.concat(orderedModifiers.filter(function (modifier) {
      return modifier.phase === phase;
    }));
  }, []);
}

function debounce(fn) {
  var pending;
  return function () {
    if (!pending) {
      pending = new Promise(function (resolve) {
        Promise.resolve().then(function () {
          pending = undefined;
          resolve(fn());
        });
      });
    }

    return pending;
  };
}

function mergeByName(modifiers) {
  var merged = modifiers.reduce(function (merged, current) {
    var existing = merged[current.name];
    merged[current.name] = existing ? Object.assign({}, existing, current, {
      options: Object.assign({}, existing.options, current.options),

```



```

    data: Object.assign({}, existing.data, current.data)
  }) : current;
  return merged;
}, {}); // IE11 does not support Object.values

return Object.keys(merged).map(function (key) {
  return merged[key];
});
}

var DEFAULT_OPTIONS = {
  placement: 'bottom',
  modifiers: [],
  strategy: 'absolute'
};

function areValidElements() {
  for (var _len = arguments.length, args = new Array(_len), _key = 0; _key < _len; _key++) {
    args[_key] = arguments[_key];
  }

  return !args.some(function (element) {
    return !(element && typeof element.getBoundingClientRect === 'function');
  });
}

function popperGenerator(generatorOptions) {
  if (generatorOptions === void 0) {
    generatorOptions = {};
  }

  var _generatorOptions = generatorOptions,
      _generatorOptions$def = _generatorOptions.defaultModifiers,
      defaultModifiers = _generatorOptions$def === void 0 ? [] : _generatorOptions$def,
      _generatorOptions$def2 = _generatorOptions.defaultOptions,
      defaultOptions = _generatorOptions$def2 === void 0 ? DEFAULT_OPTIONS :
  _generatorOptions$def2;
  return function createPopper(reference, popper, options) {
    if (options === void 0) {
      options = defaultOptions;
    }

    var state = {
      placement: 'bottom',
      orderedModifiers: [],
      options: Object.assign({}, DEFAULT_OPTIONS, defaultOptions),
      modifiersData: {},
      elements: {
        reference: reference,
        popper: popper
      },
      attributes: {},
      styles: {}
    };
    var effectCleanupFns = [];
    var isDestroyed = false;
    var instance = {

```

```

state: state,
setOptions: function setOptions(setOptionsAction) {
  var options = typeof setOptionsAction === 'function' ? setOptionsAction(state.options) :
setOptionsAction;
  cleanupModifierEffects();
  state.options = Object.assign({}, defaultOptions, state.options, options);
  state.scrollParents = {
    reference: isElement(reference) ? listScrollParents(reference) : reference.contextElement ?
listScrollParents(reference.contextElement) : [],
    popper: listScrollParents/popper)
  }; // Orders the modifiers based on their dependencies and `phase`
  // properties

  var orderedModifiers = orderModifiers(mergeByName([].concat(defaultModifiers,
state.options.modifiers))); // Strip out disabled modifiers

  state.orderedModifiers = orderedModifiers.filter(function (m) {
    return m.enabled;
  }); // Validate the provided modifiers so that the consumer will get warned

  runModifierEffects();
  return instance.update();
},
// Sync update – it will always be executed, even if not necessary. This
// is useful for low frequency updates where sync behavior simplifies the
// logic.
// For high frequency updates (e.g. `resize` and `scroll` events), always
// prefer the async Popper#update method
forceUpdate: function forceUpdate() {
  if (isDestroyed) {
    return;
  }

  var _state$elements = state.elements,
      reference = _state$elements.reference,
      popper = _state$elements.popper; // Don't proceed if `reference` or `popper` are not valid
elements
  // anymore

  if (!areValidElements(reference, popper)) {

    return;
  } // Store the reference and popper rects to be read by modifiers

  state.rects = {
    reference: getCompositeRect(reference, getOffsetParent/popper), state.options.strategy ===
'fixed'),
    popper: getLayoutRect/popper)
  }; // Modifiers have the ability to reset the current update cycle. The
  // most common use case for this is the `flip` modifier changing the
  // placement, which then needs to re-run all the modifiers, because the
  // logic was previously ran for the previous placement and is therefore
  // stale/incorrect

  state.reset = false;

```

```

    state.placement = state.options.placement; // On each update cycle, the `modifiersData`
property for each modifier
    // is filled with the initial data specified by the modifier. This means
    // it doesn't persist and is fresh on each update.
    // To ensure persistent data, use `${name}#persistent`

    state.orderedModifiers.forEach(function (modifier) {
        return state.modifiersData[modifier.name] = Object.assign({}, modifier.data);
    });

    for (var index = 0; index < state.orderedModifiers.length; index++) {

        if (state.reset === true) {
            state.reset = false;
            index = -1;
            continue;
        }

        var _state$orderedModifie = state.orderedModifiers[index],
            fn = _state$orderedModifie.fn,
            _state$orderedModifie2 = _state$orderedModifie.options,
            _options = _state$orderedModifie2 === void 0 ? {} : _state$orderedModifie2,
            name = _state$orderedModifie.name;

        if (typeof fn === 'function') {
            state = fn({
                state: state,
                options: _options,
                name: name,
                instance: instance
            }) || state;
        }
    },
    // Async and optimistically optimized update – it will not be executed if
    // not necessary (debounced to run at most once-per-tick)
    update: debounce(function () {
        return new Promise(function (resolve) {
            instance.forceUpdate();
            resolve(state);
        });
    }),
    destroy: function destroy() {
        cleanupModifierEffects();
        isDestroyed = true;
    }
};

if (!areValidElements(reference, popper)) {

    return instance;
}

instance.setOptions(options).then(function (state) {
    if (!isDestroyed && options.onFirstUpdate) {
        options.onFirstUpdate(state);
    }
}

```

```

}); // Modifiers have the ability to execute arbitrary code before the first
// update cycle runs. They will be executed in the same order as the update
// cycle. This is useful when a modifier adds some persistent data that
// other modifiers need to use, but the modifier is run after the dependent
// one.

function runModifierEffects() {
  state.orderedModifiers.forEach(function (_ref3) {
    var name = _ref3.name,
        _ref3$options = _ref3.options,
        options = _ref3$options === void 0 ? {} : _ref3$options,
        effect = _ref3.effect;

    if (typeof effect === 'function') {
      var cleanupFn = effect({
        state: state,
        name: name,
        instance: instance,
        options: options
      });

      var noopFn = function noopFn() {};

      effectCleanupFns.push(cleanupFn || noopFn);
    }
  });
}

function cleanupModifierEffects() {
  effectCleanupFns.forEach(function (fn) {
    return fn();
  });
  effectCleanupFns = [];
}

return instance;
};
}

var createPopper$2 = /*#__PURE__*/popperGenerator(); // eslint-disable-next-line import/no-unused-modules

var defaultModifiers$1 = [eventListeners, popperOffsets$1, computeStyles$1, applyStyles$1];
var createPopper$1 = /*#__PURE__*/popperGenerator({
  defaultModifiers: defaultModifiers$1
}); // eslint-disable-next-line import/no-unused-modules

var defaultModifiers = [eventListeners, popperOffsets$1, computeStyles$1, applyStyles$1, offset$1,
flip$1, preventOverflow$1, arrow$1, hide$1];
var createPopper = /*#__PURE__*/popperGenerator({
  defaultModifiers: defaultModifiers
}); // eslint-disable-next-line import/no-unused-modules

const Popper = /*#__PURE__*/Object.freeze(/*#__PURE__*/Object.defineProperty({
  __proto__: null,
  popperGenerator,
  detectOverflow,
  createPopperBase: createPopper$2,

```

```

createPopper,
createPopperLite: createPopper$1,
top,
bottom,
right,
left,
auto,
basePlacements,
start,
end,
clippingParents,
viewport,
popper,
reference,
variationPlacements,
placements,
beforeRead,
read,
afterRead,
beforeMain,
main,
afterMain,
beforeWrite,
write,
afterWrite,
modifierPhases,
applyStyles: applyStyles$1,
arrow: arrow$1,
computeStyles: computeStyles$1,
eventListeners,
flip: flip$1,
hide: hide$1,
offset: offset$1,
popperOffsets: popperOffsets$1,
preventOverflow: preventOverflow$1
}, Symbol.toStringTag, { value: 'Module' }));

```

```

/**
 * -----
 * Bootstrap (v5.2.2): dropdown.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

```

```

const NAME$a = 'dropdown';
const DATA_KEY$6 = 'bs.dropdown';
const EVENT_KEY$6 = `.${DATA_KEY$6}`;
const DATA_API_KEY$3 = '.data-api';
const ESCAPE_KEY$2 = 'Escape';
const TAB_KEY$1 = 'Tab';
const ARROW_UP_KEY$1 = 'ArrowUp';
const ARROW_DOWN_KEY$1 = 'ArrowDown';
const RIGHT_MOUSE_BUTTON = 2; // MouseEvent.button value for the secondary button, usually
the right button

```

```

const EVENT_HIDE$5 = `hide${EVENT_KEY$6}`;
const EVENT_HIDDEN$5 = `hidden${EVENT_KEY$6}`;
const EVENT_SHOW$5 = `show${EVENT_KEY$6}`;
const EVENT_SHOWN$5 = `shown${EVENT_KEY$6}`;
const EVENT_CLICK_DATA_API$3 = `click${EVENT_KEY$6}${DATA_API_KEY$3}`;
const EVENT_KEYDOWN_DATA_API = `keydown${EVENT_KEY$6}${DATA_API_KEY$3}`;
const EVENT_KEYUP_DATA_API = `keyup${EVENT_KEY$6}${DATA_API_KEY$3}`;
const CLASS_NAME_SHOW$6 = 'show';
const CLASS_NAME_DROPUP = 'dropup';
const CLASS_NAME_DROPEND = 'dropend';
const CLASS_NAME_DROPSTART = 'dropstart';
const CLASS_NAME_DROPUP_CENTER = 'dropup-center';
const CLASS_NAME_DROPDOWN_CENTER = 'dropdown-center';
const SELECTOR_DATA_TOGGLE$3 = '[data-bs-toggle="dropdown"]:not(.disabled):not(:disabled)';
const SELECTOR_DATA_TOGGLE_SHOWN =
`${SELECTOR_DATA_TOGGLE$3}.${CLASS_NAME_SHOW$6}`;
const SELECTOR_MENU = '.dropdown-menu';
const SELECTOR_NAVBAR = '.navbar';
const SELECTOR_NAVBAR_NAV = '.navbar-nav';
const SELECTOR_VISIBLE_ITEMS = '.dropdown-menu .dropdown-item:not(.disabled):not(:disabled)';
const PLACEMENT_TOP = isRTL() ? 'top-end' : 'top-start';
const PLACEMENT_TOPEND = isRTL() ? 'top-start' : 'top-end';
const PLACEMENT_BOTTOM = isRTL() ? 'bottom-end' : 'bottom-start';
const PLACEMENT_BOTTOMEND = isRTL() ? 'bottom-start' : 'bottom-end';
const PLACEMENT_RIGHT = isRTL() ? 'left-start' : 'right-start';
const PLACEMENT_LEFT = isRTL() ? 'right-start' : 'left-start';
const PLACEMENT_TOPCENTER = 'top';
const PLACEMENT_BOTTOMCENTER = 'bottom';
const Default$9 = {
  autoClose: true,
  boundary: 'clippingParents',
  display: 'dynamic',
  offset: [0, 2],
  popperConfig: null,
  reference: 'toggle'
};
const DefaultType$9 = {
  autoClose: '(boolean|string)',
  boundary: '(string|element)',
  display: 'string',
  offset: '(array|string|function)',
  popperConfig: '(null|object|function)',
  reference: '(string|element|object)'
};
/**
 * Class definition
 */

```

```

class Dropdown extends BaseComponent {
  constructor(element, config) {
    super(element, config);
    this._popper = null;
    this._parent = this._element.parentNode; // dropdown wrapper
    // todo: v6 revert #37011 & change markup https://getbootstrap.com/docs/5.2/forms/input-
    group/
  }

```

```

    this._menu = SelectorEngine.next(this._element, SELECTOR_MENU)[0] ||
SelectorEngine.prev(this._element, SELECTOR_MENU)[0] ||
SelectorEngine.findOne(SELECTOR_MENU, this._parent);
    this._inNavbar = this._detectNavbar();
} // Getters


static get Default() {
    return Default$9;
}

static get DefaultType() {
    return DefaultType$9;
}

static get NAME() {
    return NAME$a;
} // Public


toggle() {
    return this._isShown() ? this.hide() : this.show();
}

show() {
    if (isDisabled(this._element) || this._isShown()) {
        return;
    }

    const relatedTarget = {
        relatedTarget: this._element
    };
    const showEvent = EventHandler.trigger(this._element, EVENT_SHOW$5, relatedTarget);

    if (showEvent.defaultPrevented) {
        return;
    }

    this._createPopper(); // If this is a touch-enabled device we add extra
    // empty mouseover listeners to the body's immediate children;
    // only needed because of broken event delegation on iOS
    // https://www.quirksmode.org/blog/archives/2014/02/mouse_event_bub.html

    if ('ontouchstart' in document.documentElement
    && !this._parent.closest(SELECTOR_NAVBAR_NAV)) {
        for (const element of [].concat(...document.body.children)) {
            EventHandler.on(element, 'mouseover', noop);
        }
    }

    this._element.focus();

    this._element.setAttribute('aria-expanded', true);

    this._menu.classList.add(CLASS_NAME_SHOW$6);

```

```

    this._element.classList.add(CLASS_NAME_SHOW$6);

    EventHandler.trigger(this._element, EVENT_SHOWN$5, relatedTarget);
  }

  hide() {
    if (isDisabled(this._element) || !this._isShown()) {
      return;
    }

    const relatedTarget = {
      relatedTarget: this._element
    };

    this._completeHide(relatedTarget);
  }

  dispose() {
    if (this._popper) {
      this._popper.destroy();
    }

    super.dispose();
  }

  update() {
    this._inNavbar = this._detectNavbar();

    if (this._popper) {
      this._popper.update();
    }
  } // Private

  _completeHide(relatedTarget) {
    const hideEvent = EventHandler.trigger(this._element, EVENT_HIDE$5, relatedTarget);

    if (hideEvent.defaultPrevented) {
      return;
    } // If this is a touch-enabled device we remove the extra
    // empty mouseover listeners we added for iOS support

    if ('ontouchstart' in document.documentElement) {
      for (const element of [].concat(...document.body.children)) {
        EventHandler.off(element, 'mouseover', noop);
      }
    }

    if (this._popper) {
      this._popper.destroy();
    }

    this._menu.classList.remove(CLASS_NAME_SHOW$6);

    this._element.classList.remove(CLASS_NAME_SHOW$6);

```



```

this._element.setAttribute('aria-expanded', 'false');

Manipulator.removeDataAttribute(this._menu, 'popper');
EventHandler.trigger(this._element, EVENT_HIDDEN$5, relatedTarget);
}

_getConfig(config) {
  config = super._getConfig(config);

  if (typeof config.reference === 'object' && !isElement$1(config.reference) && typeof
config.reference.getBoundingClientRect !== 'function') {
    // Popper virtual elements require a getBoundingClientRect method
    throw new TypeError(`${NAME$a.toUpperCase()}: Option "reference" provided type "object"
without a required "getBoundingClientRect" method.`);
  }

  return config;
}

_createPopper() {
  if (typeof Popper === 'undefined') {
    throw new TypeError('Bootstrap\'s dropdowns require Popper (https://popper.js.org)');
  }

  let referenceElement = this._element;

  if (this._config.reference === 'parent') {
    referenceElement = this._parent;
  } else if (isElement$1(this._config.reference)) {
    referenceElement = getElement(this._config.reference);
  } else if (typeof this._config.reference === 'object') {
    referenceElement = this._config.reference;
  }

  const popperConfig = this._getPopperConfig();

  this._popper = createPopper(referenceElement, this._menu, popperConfig);
}

_isShown() {
  return this._menu.classList.contains(CLASS_NAME_SHOW$6);
}

_getPlacement() {
  const parentDropdown = this._parent;

  if (parentDropdown.classList.contains(CLASS_NAME_DROPEND)) {
    return PLACEMENT_RIGHT;
  }

  if (parentDropdown.classList.contains(CLASS_NAME_DROPSTART)) {
    return PLACEMENT_LEFT;
  }

  if (parentDropdown.classList.contains(CLASS_NAME_DROPUP_CENTER)) {
    return PLACEMENT_TOPCENTER;
  }
}

```

```

    if (parentDropdown.classList.contains(CLASS_NAME_DROPDOWN_CENTER)) {
      return PLACEMENT_BOTTOMCENTER;
    } // We need to trim the value because custom properties can also include spaces

    const isEnd = getComputedStyle(this._menu).getPropertyValue('--bs-position').trim() === 'end';

    if (parentDropdown.classList.contains(CLASS_NAME_DROPUP)) {
      return isEnd ? PLACEMENT_TOPEND : PLACEMENT_TOP;
    }

    return isEnd ? PLACEMENT_BOTTOMEND : PLACEMENT_BOTTOM;
  }

  _detectNavbar() {
    return this._element.closest(SELECTOR_NAVBAR) !== null;
  }

  _getOffset() {
    const {
      offset
    } = this._config;

    if (typeof offset === 'string') {
      return offset.split(',').map(value => Number.parseInt(value, 10));
    }

    if (typeof offset === 'function') {
      return popperData => offset(popperData, this._element);
    }

    return offset;
  }

  _getPopperConfig() {
    const defaultBsPopperConfig = {
      placement: this._getPlacement(),
      modifiers: [{
        name: 'preventOverflow',
        options: {
          boundary: this._config.boundary
        }
      }],
    };
    return [
      {
        name: 'offset',
        options: {
          offset: this._getOffset()
        }
      }
    ];
  }

  // Disable Popper if we have a static display or Dropdown is in Navbar
  if (this._inNavbar || this._config.display === 'static') {
    Manipulator.setDataAttribute(this._menu, 'popper', 'static'); // todo:v6 remove

    defaultBsPopperConfig.modifiers = [{
      name: 'applyStyles',
      enabled: false
    }];
  }

```

```

    });
  }

  return { ...defaultBsPopperConfig,
    ...(typeof this._config.popperConfig === 'function' ?
this._config.popperConfig(defaultBsPopperConfig) : this._config.popperConfig)
  };
}

_selectMenuItem({
  key,
  target
}) {
  const items = SelectorEngine.find(SELECTOR_VISIBLE_ITEMS, this._menu).filter(element =>
isVisible(element));

  if (!items.length) {
    return;
  } // if target isn't included in items (e.g. when expanding the dropdown)
  // allow cycling to get the last item in case key equals ARROW_UP_KEY

  getNextActiveElement(items, target, key ===
ARROW_DOWN_KEY$1, !items.includes(target)).focus();
} // Static

static jQueryInterface(config) {
  return this.each(function () {
    const data = Dropdown.getOrCreateInstance(this, config);

    if (typeof config !== 'string') {
      return;
    }

    if (typeof data[config] === 'undefined') {
      throw new TypeError(`No method named "${config}"`);
    }

    data[config]();
  });
}

static clearMenus(event) {
  if (event.button === RIGHT_MOUSE_BUTTON || event.type === 'keyup' && event.key !==
TAB_KEY$1) {
    return;
  }

  const openToggles = SelectorEngine.find(SELECTOR_DATA_TOGGLE_SHOWN);

  for (const toggle of openToggles) {
    const context = Dropdown.getInstance(toggle);

    if (!context || context._config.autoClose === false) {
      continue;
    }

```

```

const composedPath = event.composedPath();
const isMenuTarget = composedPath.includes(context._menu);

if (composedPath.includes(context._element) || context._config.autoClose === 'inside'
&& !isMenuTarget || context._config.autoClose === 'outside' && isMenuTarget) {
  continue;
} // Tab navigation through the dropdown menu or events from contained inputs shouldn't close
the menu

if (context._menu.contains(event.target) && (event.type === 'keyup' && event.key ===
TAB_KEY$1 || /input|select|option|textarea|form/i.test(event.target.tagName))) {
  continue;
}

const relatedTarget = {
  relatedTarget: context._element
};

if (event.type === 'click') {
  relatedTarget.clickEvent = event;
}

context._completeHide(relatedTarget);
}
}

static dataApiKeyDownHandler(event) {
  // If not an UP | DOWN | ESCAPE key => not a dropdown command
  // If input/textarea && if key is other than ESCAPE => not a dropdown command
  const isInput = /input|textarea/i.test(event.target.tagName);
  const isEscapeEvent = event.key === ESCAPE_KEY$2;
  const isUpOrDownEvent = [ARROW_UP_KEY$1, ARROW_DOWN_KEY$1].includes(event.key);

  if (!isUpOrDownEvent && !isEscapeEvent) {
    return;
  }

  if (isInput && !isEscapeEvent) {
    return;
  }

  event.preventDefault(); // todo: v6 revert #37011 & change markup
https://getbootstrap.com/docs/5.2/forms/input-group/

  const getToggleButton = this.matches(SELECTOR_DATA_TOGGLE$3) ? this :
  SelectorEngine.prev(this, SELECTOR_DATA_TOGGLE$3)[0] || SelectorEngine.next(this,
  SELECTOR_DATA_TOGGLE$3)[0] || SelectorEngine.findOne(SELECTOR_DATA_TOGGLE$3,
  event.delegateTarget.parentNode);
  const instance = Dropdown.getOrCreateInstance(getToggleButton);

  if (isUpOrDownEvent) {
    event.stopPropagation();
    instance.show();

    instance._selectMenuItem(event);
  }
}

```

```

        return;
    }

    if (instance._isShown()) {
        // else is escape and we check if it is shown
        event.stopPropagation();
        instance.hide();
        getToggleButton.focus();
    }
}

}
/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_KEYDOWN_DATA_API, SELECTOR_DATA_TOGGLE$3,
Dropdown.dataApiKeydownHandler);
EventHandler.on(document, EVENT_KEYDOWN_DATA_API, SELECTOR_MENU,
Dropdown.dataApiKeydownHandler);
EventHandler.on(document, EVENT_CLICK_DATA_API$3, Dropdown.clearMenus);
EventHandler.on(document, EVENT_KEYUP_DATA_API, Dropdown.clearMenus);
EventHandler.on(document, EVENT_CLICK_DATA_API$3, SELECTOR_DATA_TOGGLE$3, function
(event) {
    event.preventDefault();
    Dropdown.getOrCreateInstance(this).toggle();
});
/**
 * jQuery
 */

defineJQueryPlugin(Dropdown);

/**
 * -----
 * Bootstrap (v5.2.2): util/scrollBar.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const SELECTOR_FIXED_CONTENT = '.fixed-top, .fixed-bottom, .is-fixed, .sticky-top';
const SELECTOR_STICKY_CONTENT = '.sticky-top';
const PROPERTY_PADDING = 'padding-right';
const PROPERTY_MARGIN = 'margin-right';
/**
 * Class definition
 */

class ScrollBarHelper {
    constructor() {
        this._element = document.body;
    } // Public

```

```

getWidth() {
  // https://developer.mozilla.org/en-US/docs/Web/API/Window/innerWidth#usage_notes
  const documentWidth = document.documentElement.clientWidth;
  return Math.abs(window.innerWidth - documentWidth);
}

hide() {
  const width = this.getWidth();

  this._disableOverflow(); // give padding to element to balance the hidden scrollbar width

  this._setElementAttributes(this._element, PROPERTY_PADDING, calculatedValue =>
    calculatedValue + width); // trick: We adjust positive paddingRight and negative marginRight to sticky-
    top elements to keep showing fullwidth

  this._setElementAttributes(SELECTOR_FIXED_CONTENT, PROPERTY_PADDING, calculatedValue =>
    calculatedValue + width);

  this._setElementAttributes(SELECTOR_STICKY_CONTENT, PROPERTY_MARGIN, calculatedValue =>
    calculatedValue - width);
}

reset() {
  this._resetElementAttributes(this._element, 'overflow');

  this._resetElementAttributes(this._element, PROPERTY_PADDING);

  this._resetElementAttributes(SELECTOR_FIXED_CONTENT, PROPERTY_PADDING);

  this._resetElementAttributes(SELECTOR_STICKY_CONTENT, PROPERTY_MARGIN);
}

isOverflowing() {
  return this.getWidth() > 0;
} // Private

_disableOverflow() {
  this._saveInitialAttribute(this._element, 'overflow');

  this._element.style.overflow = 'hidden';
}

_setElementAttributes(selector, styleProperty, callback) {
  const scrollbarWidth = this.getWidth();

  const manipulationCallBack = element => {
    if (element !== this._element && window.innerWidth > element.clientWidth + scrollbarWidth) {
      return;
    }

    this._saveInitialAttribute(element, styleProperty);
  }

```

```

    const calculatedValue = window.getComputedStyle(element).getPropertyValue(styleProperty);
    element.style.setProperty(styleProperty, `${callback(Number.parseFloat(calculatedValue))}px`);
  };

  this._applyManipulationCallback(selector, manipulationCallBack);
}

_saveInitialAttribute(element, styleProperty) {
  const actualValue = element.style.getPropertyValue(styleProperty);

  if (actualValue) {
    Manipulator.setDataAttribute(element, styleProperty, actualValue);
  }
}

_resetElementAttributes(selector, styleProperty) {
  const manipulationCallBack = element => {
    const value = Manipulator.getDataAttribute(element, styleProperty); // We only want to remove
    the property if the value is `null`; the value can also be zero

    if (value === null) {
      element.style.removeProperty(styleProperty);
      return;
    }

    Manipulator.removeDataAttribute(element, styleProperty);
    element.style.setProperty(styleProperty, value);
  };

  this._applyManipulationCallback(selector, manipulationCallBack);
}

_applyManipulationCallback(selector, callBack) {
  if (isElement$1(selector)) {
    callBack(selector);
    return;
  }

  for (const sel of SelectorEngine.find(selector, this._element)) {
    callBack(sel);
  }
}

}

/**
 * -----
 * Bootstrap (v5.2.2): util/backdrop.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$9 = 'backdrop';
const CLASS_NAME_FADE$4 = 'fade';

```

```

const CLASS_NAME_SHOW$5 = 'show';
const EVENT_MOUSEDOWN = `mousedown.bs.${NAME$9}`;
const Default$8 = {
  className: 'modal-backdrop',
  clickCallback: null,
  isAnimated: false,
  isVisible: true,
  // if false, we use the backdrop helper without adding any element to the dom
  rootElement: 'body' // give the choice to place backdrop under different elements
};
const DefaultType$8 = {
  className: 'string',
  clickCallback: '(function| null)',
  isAnimated: 'boolean',
  isVisible: 'boolean',
  rootElement: '(element|string)'
};
/**
 * Class definition
 */
class Backdrop extends Config {
  constructor(config) {
    super();
    this._config = this._getConfig(config);
    this._isAppended = false;
    this._element = null;
  } // Getters

  static get Default() {
    return Default$8;
  }

  static get DefaultType() {
    return DefaultType$8;
  }

  static get NAME() {
    return NAME$9;
  } // Public

  show(callback) {
    if (!this._config.isVisible) {
      execute(callback);
      return;
    }
  }

  this._append();

  const element = this._getElement();

  if (this._config.isAnimated) {
    reflow(element);
  }

```



```

element.classList.add(CLASS_NAME_SHOW$5);

this._emulateAnimation(() => {
  execute(callback);
});
}

hide(callback) {
  if (!this._config.isVisible) {
    execute(callback);
    return;
  }

  this._getElement().classList.remove(CLASS_NAME_SHOW$5);

  this._emulateAnimation(() => {
    this.dispose();
    execute(callback);
  });
}

dispose() {
  if (!this._isAppended) {
    return;
  }

  EventHandler.off(this._element, EVENT_MOUSEDOWN);

  this._element.remove();

  this._isAppended = false;
} // Private

_getElement() {
  if (!this._element) {
    const backdrop = document.createElement('div');
    backdrop.className = this._config.className;

    if (this._config.isAnimated) {
      backdrop.classList.add(CLASS_NAME_FADE$4);
    }

    this._element = backdrop;
  }

  return this._element;
}

_configAfterMerge(config) {
  // use getElement() with the default "body" to get a fresh Element on each instantiation
  config.rootElement = getElement(config.rootElement);
  return config;
}

_append() {

```

```

    if (this._isAppended) {
        return;
    }

    const element = this._getElement();

    this._config.rootElement.append(element);

    EventHandler.on(element, EVENT_MOUSEDOWN, () => {
        execute(this._config.clickCallback);
    });
    this._isAppended = true;
}

_emulateAnimation(callback) {
    executeAfterTransition(callback, this._getElement(), this._config.isAnimated);
}

}

/**
 * -----
 * Bootstrap (v5.2.2): util/focustrap.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$8 = 'focustrap';
const DATA_KEY$5 = 'bs.focustrap';
const EVENT_KEY$5 = `.${DATA_KEY$5}`;
const EVENT_FOCUSIN$2 = `focusin${EVENT_KEY$5}`;
const EVENT_KEYDOWN_TAB = `keydown.tab${EVENT_KEY$5}`;
const TAB_KEY = 'Tab';
const TAB_NAV_FORWARD = 'forward';
const TAB_NAV_BACKWARD = 'backward';
const Default$7 = {
    autofocus: true,
    trapElement: null // The element to trap focus inside of
};

const DefaultType$7 = {
    autofocus: 'boolean',
    trapElement: 'element'
};

/**
 * Class definition
 */

class FocusTrap extends Config {
    constructor(config) {
        super();
        this._config = this._getConfig(config);
        this._isActive = false;
        this._lastTabNavDirection = null;
    }

```

```

} // Getters

static get Default() {
  return Default$7;
}

static get DefaultType() {
  return DefaultType$7;
}

static get NAME() {
  return NAME$8;
} // Public

activate() {
  if (this._isActive) {
    return;
  }

  if (this._config.autofocus) {
    this._config.trapElement.focus();
  }

  EventHandler.off(document, EVENT_KEY$5); // guard against infinite focus loop

  EventHandler.on(document, EVENT_FOCUSIN$2, event => this._handleFocusin(event));
  EventHandler.on(document, EVENT_KEYDOWN_TAB, event => this._handleKeydown(event));
  this._isActive = true;
}

deactivate() {
  if (!this._isActive) {
    return;
  }

  this._isActive = false;
  EventHandler.off(document, EVENT_KEY$5);
} // Private

_handleFocusin(event) {
  const {
    trapElement
  } = this._config;

  if (event.target === document || event.target === trapElement ||
  trapElement.contains(event.target)) {
    return;
  }

  const elements = SelectorEngine.focusableChildren(trapElement);

  if (elements.length === 0) {
    trapElement.focus();
  } else if (this._lastTabNavDirection === TAB_NAV_BACKWARD) {

```

```

        elements[elements.length - 1].focus();
    } else {
        elements[0].focus();
    }
}

_handleKeydown(event) {
    if (event.key !== TAB_KEY) {
        return;
    }

    this._lastTabNavDirection = event.shiftKey ? TAB_NAV_BACKWARD : TAB_NAV_FORWARD;
}

}

/**
 * -----
 * Bootstrap (v5.2.2): modal.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$7 = 'modal';
const DATA_KEY$4 = 'bs.modal';
const EVENT_KEY$4 = `.${DATA_KEY$4}`;
const DATA_API_KEY$2 = '.data-api';
const ESCAPE_KEY$1 = 'Escape';
const EVENT_HIDE$4 = `hide${EVENT_KEY$4}`;
const EVENT_HIDE_PREVENTED$1 = `hidePrevented${EVENT_KEY$4}`;
const EVENT_HIDDEN$4 = `hidden${EVENT_KEY$4}`;
const EVENT_SHOW$4 = `show${EVENT_KEY$4}`;
const EVENT_SHOWN$4 = `shown${EVENT_KEY$4}`;
const EVENT_RESIZE$1 = `resize${EVENT_KEY$4}`;
const EVENT_CLICK_DISMISS = `click.dismiss${EVENT_KEY$4}`;
const EVENT_MOUSEDOWN_DISMISS = `mousedown.dismiss${EVENT_KEY$4}`;
const EVENT_KEYDOWN_DISMISS$1 = `keydown.dismiss${EVENT_KEY$4}`;
const EVENT_CLICK_DATA_API$2 = `click${EVENT_KEY$4}${DATA_API_KEY$2}`;
const CLASS_NAME_OPEN = 'modal-open';
const CLASS_NAME_FADE$3 = 'fade';
const CLASS_NAME_SHOW$4 = 'show';
const CLASS_NAME_STATIC = 'modal-static';
const OPEN_SELECTOR$1 = '.modal.show';
const SELECTOR_DIALOG = '.modal-dialog';
const SELECTOR_MODAL_BODY = '.modal-body';
const SELECTOR_DATA_TOGGLE$2 = '[data-bs-toggle="modal"]';
const Default$6 = {
    backdrop: true,
    focus: true,
    keyboard: true
};
const DefaultType$6 = {
    backdrop: '(boolean|string)',
    focus: 'boolean',

```

```

    keyboard: 'boolean'
};
/**
 * Class definition
 */

class Modal extends BaseComponent {
  constructor(element, config) {
    super(element, config);
    this._dialog = SelectorEngine.findOne(SELECTOR_DIALOG, this._element);
    this._backdrop = this._initializeBackDrop();
    this._focustrap = this._initializeFocusTrap();
    this._isShown = false;
    this._isTransitioning = false;
    this._scrollBar = new ScrollBarHelper();

    this._addEventListeners();
  } // Getters

  static get Default() {
    return Default$6;
  }

  static get DefaultType() {
    return DefaultType$6;
  }

  static get NAME() {
    return NAME$7;
  } // Public

  toggle(relatedTarget) {
    return this._isShown ? this.hide() : this.show(relatedTarget);
  }

  show(relatedTarget) {
    if (this._isShown || this._isTransitioning) {
      return;
    }

    const showEvent = EventHandler.trigger(this._element, EVENT_SHOW$4, {
      relatedTarget
    });

    if (showEvent.defaultPrevented) {
      return;
    }

    this._isShown = true;
    this._isTransitioning = true;

    this._scrollBar.hide();

    document.body.classList.add(CLASS_NAME_OPEN);

```

```

    this._adjustDialog();

    this._backdrop.show(() => this._showElement(relatedTarget));
  }

  hide() {
    if (!this._isShown || this._isTransitioning) {
      return;
    }

    const hideEvent = EventHandler.trigger(this._element, EVENT_HIDE$4);

    if (hideEvent.defaultPrevented) {
      return;
    }

    this._isShown = false;
    this._isTransitioning = true;

    this._focustrap.deactivate();

    this._element.classList.remove(CLASS_NAME_SHOW$4);

    this._queueCallback(() => this._hideModal(), this._element, this._isAnimated());
  }

  dispose() {
    for (const htmlElement of [window, this._dialog]) {
      EventHandler.off(htmlElement, EVENT_KEY$4);
    }

    this._backdrop.dispose();

    this._focustrap.deactivate();

    super.dispose();
  }

  handleUpdate() {
    this._adjustDialog();
  } // Private

  _initializeBackDrop() {
    return new Backdrop({
      isVisible: Boolean(this._config.backdrop),
      // 'static' option will be translated to true, and booleans will keep their value,
      isAnimated: this._isAnimated()
    });
  }

  _initializeFocusTrap() {
    return new FocusTrap({
      trapElement: this._element
    });
  }

```

```

_showElement(relatedTarget) {
  // try to append dynamic modal
  if (!document.body.contains(this._element)) {
    document.body.append(this._element);
  }

  this._element.style.display = 'block';

  this._element.removeAttribute('aria-hidden');

  this._element.setAttribute('aria-modal', true);

  this._element.setAttribute('role', 'dialog');

  this._element.scrollTop = 0;
  const modalBody = SelectorEngine.findOne(SELECTOR_MODAL_BODY, this._dialog);

  if (modalBody) {
    modalBody.scrollTop = 0;
  }

  reflow(this._element);

  this._element.classList.add(CLASS_NAME_SHOW$4);

  const transitionComplete = () => {
    if (this._config.focus) {
      this._focustrap.activate();
    }

    this._isTransitioning = false;
    EventHandler.trigger(this._element, EVENT_SHOWN$4, {
      relatedTarget
    });
  };

  this._queueCallback(transitionComplete, this._dialog, this._isAnimated());
}

_addEventListeners() {
  EventHandler.on(this._element, EVENT_KEYDOWN_DISMISS$1, event => {
    if (event.key !== ESCAPE_KEY$1) {
      return;
    }

    if (this._config.keyboard) {
      event.preventDefault();
      this.hide();
      return;
    }

    this._triggerBackdropTransition();
  });
  EventHandler.on(window, EVENT_RESIZE$1, () => {
    if (this._isShown && !this._isTransitioning) {
      this._adjustDialog();
    }
  })
}

```

```

});
EventHandler.on(this._element, EVENT_MOUSEDOWN_DISMISS, event => {
  // a bad trick to segregate clicks that may start inside dialog but end outside, and avoid listen to
  scrollbar clicks
  EventHandler.one(this._element, EVENT_CLICK_DISMISS, event2 => {
    if (this._element !== event.target || this._element !== event2.target) {
      return;
    }

    if (this._config.backdrop === 'static') {
      this._triggerBackdropTransition();

      return;
    }

    if (this._config.backdrop) {
      this.hide();
    }
  });
});

_hideModal() {
  this._element.style.display = 'none';

  this._element.setAttribute('aria-hidden', true);

  this._element.removeAttribute('aria-modal');

  this._element.removeAttribute('role');

  this._isTransitioning = false;

  this._backdrop.hide(() => {
    document.body.classList.remove(CLASS_NAME_OPEN);

    this._resetAdjustments();

    this._scrollBar.reset();

    EventHandler.trigger(this._element, EVENT_HIDDEN$4);
  });
}

_isAnimated() {
  return this._element.classList.contains(CLASS_NAME_FADE$3);
}

_triggerBackdropTransition() {
  const hideEvent = EventHandler.trigger(this._element, EVENT_HIDE_PREVENTED$1);

  if (hideEvent.defaultPrevented) {
    return;
  }

  const isModalOverflowing = this._element.scrollHeight >
  document.documentElement.clientHeight;

```



```
const initialOverflowY = this._element.style.overflowY; // return if the following background transition hasn't yet completed
```

```
if (initialOverflowY === 'hidden' || this._element.classList.contains(CLASS_NAME_STATIC)) {  
  return;  
}
```

```
if (!isModalOverflowing) {  
  this._element.style.overflowY = 'hidden';  
}
```

```
this._element.classList.add(CLASS_NAME_STATIC);
```

```
this._queueCallback(() => {  
  this._element.classList.remove(CLASS_NAME_STATIC);
```

```
  this._queueCallback(() => {  
    this._element.style.overflowY = initialOverflowY;  
  }, this._dialog);  
}, this._dialog);
```

```
this._element.focus();
```

```
}  
/**
```

```
 * The following methods are used to handle overflowing modals  
 */
```

```
_adjustDialog() {  
  const isModalOverflowing = this._element.scrollHeight >  
document.documentElement.clientHeight;
```

```
const scrollbarWidth = this._scrollBar.getWidth();
```

```
const isBodyOverflowing = scrollbarWidth > 0;
```

```
if (isBodyOverflowing && !isModalOverflowing) {  
  const property = isRTL() ? 'paddingLeft' : 'paddingRight';  
  this._element.style[property] = `${scrollbarWidth}px`;  
}
```

```
if (!isBodyOverflowing && isModalOverflowing) {  
  const property = isRTL() ? 'paddingRight' : 'paddingLeft';  
  this._element.style[property] = `${scrollbarWidth}px`;  
}  
}
```

```
_resetAdjustments() {  
  this._element.style.paddingLeft = "";  
  this._element.style.paddingRight = "";  
} // Static
```

```
static jQueryInterface(config, relatedTarget) {  
  return this.each(function () {  
    const data = Modal.getOrCreateInstance(this, config);
```

```

    if (typeof config !== 'string') {
      return;
    }

    if (typeof data[config] === 'undefined') {
      throw new TypeError(`No method named "${config}"`);
    }

    data[config](relatedTarget);
  });
}

/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API$2, SELECTOR_DATA_TOGGLE$2, function
(event) {
  const target = getElementFromSelector(this);

  if (['A', 'AREA'].includes(this.tagName)) {
    event.preventDefault();
  }

  EventHandler.one(target, EVENT_SHOW$4, showEvent => {
    if (showEvent.defaultPrevented) {
      // only register focus restorer if modal will actually get shown
      return;
    }
  })

  EventHandler.one(target, EVENT_HIDDEN$4, () => {
    if (isVisible(this)) {
      this.focus();
    }
  });
}); // avoid conflict when clicking modal toggler while another one is open

const alreadyOpen = SelectorEngine.findOne(OPEN_SELECTOR$1);

if (alreadyOpen) {
  Modal.getInstance(alreadyOpen).hide();
}

const data = Modal.getOrCreateInstance(target);
data.toggle(this);
});
enableDismissTrigger(Modal);
/**
 * jQuery
 */

defineJQueryPlugin(Modal);

/**
 * -----

```

```

* Bootstrap (v5.2.2): offcanvas.js
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
* -----
*/
/**
* Constants
*/

const NAME$6 = 'offcanvas';
const DATA_KEY$3 = 'bs.offcanvas';
const EVENT_KEY$3 = `.${DATA_KEY$3}`;
const DATA_API_KEY$1 = '.data-api';
const EVENT_LOAD_DATA_API$2 = `load${EVENT_KEY$3}${DATA_API_KEY$1}`;
const ESCAPE_KEY = 'Escape';
const CLASS_NAME_SHOW$3 = 'show';
const CLASS_NAME_SHOWING$1 = 'showing';
const CLASS_NAME_HIDING = 'hiding';
const CLASS_NAME_BACKDROP = 'offcanvas-backdrop';
const OPEN_SELECTOR = '.offcanvas.show';
const EVENT_SHOW$3 = `show${EVENT_KEY$3}`;
const EVENT_SHOWN$3 = `shown${EVENT_KEY$3}`;
const EVENT_HIDE$3 = `hide${EVENT_KEY$3}`;
const EVENT_HIDE_PREVENTED = `hidePrevented${EVENT_KEY$3}`;
const EVENT_HIDDEN$3 = `hidden${EVENT_KEY$3}`;
const EVENT_RESIZE = `resize${EVENT_KEY$3}`;
const EVENT_CLICK_DATA_API$1 = `click${EVENT_KEY$3}${DATA_API_KEY$1}`;
const EVENT_KEYDOWN_DISMISS = `keydown.dismiss${EVENT_KEY$3}`;
const SELECTOR_DATA_TOGGLE$1 = '[data-bs-toggle="offcanvas"]';
const Default$5 = {
  backdrop: true,
  keyboard: true,
  scroll: false
};
const DefaultType$5 = {
  backdrop: '(boolean|string)',
  keyboard: 'boolean',
  scroll: 'boolean'
};
/**
* Class definition
*/

class Offcanvas extends BaseComponent {
  constructor(element, config) {
    super(element, config);
    this._isShown = false;
    this._backdrop = this._initializeBackDrop();
    this._focustrap = this._initializeFocusTrap();

    this._addEventListeners();
  } // Getters

  static get Default() {
    return Default$5;
  }
}

```

```

static get DefaultType() {
  return DefaultType$5;
}

static get NAME() {
  return NAME$6;
} // Public

toggle(relatedTarget) {
  return this._isShown ? this.hide() : this.show(relatedTarget);
}

show(relatedTarget) {
  if (this._isShown) {
    return;
  }

  const showEvent = EventHandler.trigger(this._element, EVENT_SHOW$3, {
    relatedTarget
  });

  if (showEvent.defaultPrevented) {
    return;
  }

  this._isShown = true;

  this._backdrop.show();

  if (!this._config.scroll) {
    new ScrollBarHelper().hide();
  }

  this._element.setAttribute('aria-modal', true);

  this._element.setAttribute('role', 'dialog');

  this._element.classList.add(CLASS_NAME_SHOWING$1);

  const completeCallBack = () => {
    if (!this._config.scroll || this._config.backdrop) {
      this._focustrap.activate();
    }
  }

  this._element.classList.add(CLASS_NAME_SHOW$3);

  this._element.classList.remove(CLASS_NAME_SHOWING$1);

  EventHandler.trigger(this._element, EVENT_SHOWN$3, {
    relatedTarget
  });
};

this._queueCallback(completeCallBack, this._element, true);
}

```

```

hide() {
  if (!this._isShown) {
    return;
  }

  const hideEvent = EventHandler.trigger(this._element, EVENT_HIDE$3);

  if (hideEvent.defaultPrevented) {
    return;
  }

  this._focustrap.deactivate();

  this._element.blur();

  this._isShown = false;

  this._element.classList.add(CLASS_NAME_HIDING);

  this._backdrop.hide();

  const completeCallback = () => {
    this._element.classList.remove(CLASS_NAME_SHOW$3, CLASS_NAME_HIDING);

    this._element.removeAttribute('aria-modal');

    this._element.removeAttribute('role');

    if (!this._config.scroll) {
      new ScrollBarHelper().reset();
    }

    EventHandler.trigger(this._element, EVENT_HIDDEN$3);
  };

  this._queueCallback(completeCallback, this._element, true);
}

dispose() {
  this._backdrop.dispose();

  this._focustrap.deactivate();

  super.dispose();
} // Private

_initializeBackDrop() {
  const clickCallback = () => {
    if (this._config.backdrop === 'static') {
      EventHandler.trigger(this._element, EVENT_HIDE_PREVENTED);
      return;
    }
  }

  this.hide();
}; // 'static' option will be translated to true, and booleans will keep their value

```

```

const isVisible = Boolean(this._config.backdrop);
return new Backdrop({
  className: CLASS_NAME_BACKDROP,
  isVisible,
  isAnimated: true,
  rootElement: this._element.parentNode,
  clickCallback: isVisible ? clickCallback : null
});
}

_initializeFocusTrap() {
  return new FocusTrap({
    trapElement: this._element
  });
}

_addEventListeners() {
  EventHandler.on(this._element, EVENT_KEYDOWN_DISMISS, event => {
    if (event.key !== ESCAPE_KEY) {
      return;
    }

    if (!this._config.keyboard) {
      EventHandler.trigger(this._element, EVENT_HIDE_PREVENTED);
      return;
    }

    this.hide();
  });
} // Static

static jQueryInterface(config) {
  return this.each(function () {
    const data = Offcanvas.getOrCreateInstance(this, config);

    if (typeof config !== 'string') {
      return;
    }

    if (data[config] === undefined || config.startsWith('_') || config === 'constructor') {
      throw new TypeError(`No method named "${config}"`);
    }

    data[config](this);
  });
}

/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API$1, SELECTOR_DATA_TOGGLE$1, function
(event) {

```

```

const target = getElementFromSelector(this);

if (['A', 'AREA'].includes(this.tagName)) {
  event.preventDefault();
}

if (isDisabled(this)) {
  return;
}

EventHandler.one(target, EVENT_HIDDEN$3, () => {
  // focus on trigger when it is closed
  if (isVisible(this)) {
    this.focus();
  }
}); // avoid conflict when clicking a toggler of an offcanvas, while another is open

const alreadyOpen = SelectorEngine.findOne(OPEN_SELECTOR);

if (alreadyOpen && alreadyOpen !== target) {
  Offcanvas.getInstance(alreadyOpen).hide();
}

const data = Offcanvas.getOrCreateInstance(target);
data.toggle(this);
});
EventHandler.on(window, EVENT_LOAD_DATA_API$2, () => {
  for (const selector of SelectorEngine.find(OPEN_SELECTOR)) {
    Offcanvas.getOrCreateInstance(selector).show();
  }
});
EventHandler.on(window, EVENT_RESIZE, () => {
  for (const element of SelectorEngine.find('[aria-modal][class*=show][class*=offcanvas-]')) {
    if (getComputedStyle(element).position !== 'fixed') {
      Offcanvas.getOrCreateInstance(element).hide();
    }
  }
});
enableDismissTrigger(Offcanvas);
/**
 * jQuery
 */

defineJQueryPlugin(Offcanvas);

/**
 * -----
 * Bootstrap (v5.2.2): util/sanitizer.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
const uriAttributes = new Set(['background', 'cite', 'href', 'itemtype', 'longdesc', 'poster', 'src', 'xlink:href']);
const ARIA_ATTRIBUTE_PATTERN = /^aria-[\w-]*$/i;
/**
 * A pattern that recognizes a commonly useful subset of URLs that are safe.
 */

```

```

* Shout-out to Angular
https://github.com/angular/angular/blob/12.2.x/packages/core/src/sanitization/url_sanitizer.ts
*/

const SAFE_URL_PATTERN = /^(?:(?:https?|mailto|ftp|tel|file|sms):|^[^#&/:?]*(?:[#/?]|$))/i;
/**
 * A pattern that matches safe data URLs. Only matches image, video and audio types.
 *
 * Shout-out to Angular
https://github.com/angular/angular/blob/12.2.x/packages/core/src/sanitization/url_sanitizer.ts
*/

const DATA_URL_PATTERN =
/^data:(?:image\/(?:bmp|gif|jpeg|jpg|png|tiff|webp)|video\/(?:mpeg|mp4|ogg|webm)|audio\/(?:
mp3|oga|ogg|opus));base64,[\d+/a-z]+=*$/i;

const allowedAttribute = (attribute, allowedAttributeList) => {
  const attributeName = attribute.nodeName.toLowerCase();

  if (allowedAttributeList.includes(attributeName)) {
    if (uriAttributes.has(attributeName)) {
      return Boolean(SAFE_URL_PATTERN.test(attribute.nodeValue) ||
DATA_URL_PATTERN.test(attribute.nodeValue));
    }
  }

  return true;
} // Check if a regular expression validates the attribute.

return allowedAttributeList.filter(attributeRegex => attributeRegex instanceof RegExp).some(regex
=> regex.test(attributeName));
};

const DefaultAllowlist = {
  // Global attributes allowed on any supplied element below.
  '*': ['class', 'dir', 'id', 'lang', 'role', ARIA_ATTRIBUTE_PATTERN],
  a: ['target', 'href', 'title', 'rel'],
  area: [],
  b: [],
  br: [],
  col: [],
  code: [],
  div: [],
  em: [],
  hr: [],
  h1: [],
  h2: [],
  h3: [],
  h4: [],
  h5: [],
  h6: [],
  i: [],
  img: ['src', 'srcset', 'alt', 'title', 'width', 'height'],
  li: [],
  ol: [],
  p: [],
  pre: [],

```



```

s: [],
small: [],
span: [],
sub: [],
sup: [],
strong: [],
u: [],
ul: []
};
function sanitizeHtml(unsafeHtml, allowList, sanitizeFunction) {
  if (!unsafeHtml.length) {
    return unsafeHtml;
  }

  if (sanitizeFunction && typeof sanitizeFunction === 'function') {
    return sanitizeFunction(unsafeHtml);
  }

  const domParser = new window.DOMParser();
  const createdDocument = domParser.parseFromString(unsafeHtml, 'text/html');
  const elements = [].concat(...createdDocument.body.querySelectorAll('*'));

  for (const element of elements) {
    const elementName = element.nodeName.toLowerCase();

    if (!Object.keys(allowList).includes(elementName)) {
      element.remove();
      continue;
    }

    const attributeList = [].concat(...element.attributes);
    const allowedAttributes = [].concat(allowList['*'] || [], allowList[elementName] || []);

    for (const attribute of attributeList) {
      if (!allowedAttribute(attribute, allowedAttributes)) {
        element.removeAttribute(attribute.nodeName);
      }
    }
  }

  return createdDocument.body.innerHTML;
}

/**
 * -----
 * Bootstrap (v5.2.2): util/template-factory.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$5 = 'TemplateFactory';
const Default$4 = {
  allowList: DefaultAllowlist,
  content: {},

```

```

    // { selector : text , selector2 : text2 , }
    extraClass: "",
    html: false,
    sanitize: true,
    sanitizeFn: null,
    template: '<div></div>'
  };
  const DefaultType$4 = {
    allowList: 'object',
    content: 'object',
    extraClass: '(string|function)',
    html: 'boolean',
    sanitize: 'boolean',
    sanitizeFn: '(null|function)',
    template: 'string'
  };
  const DefaultContentType = {
    entry: '(string|element|function|null)',
    selector: '(string|element)'
  };
  /**
   * Class definition
   */

  class TemplateFactory extends Config {
    constructor(config) {
      super();
      this._config = this._getConfig(config);
    } // Getters

    static get Default() {
      return Default$4;
    }

    static get DefaultType() {
      return DefaultType$4;
    }

    static get NAME() {
      return NAME$5;
    } // Public

    getContent() {
      return Object.values(this._config.content).map(config =>
        this._resolvePossibleFunction(config)).filter(Boolean);
    }

    hasContent() {
      return this.getContent().length > 0;
    }

    changeContent(content) {
      this._checkContent(content);

      this._config.content = { ...this._config.content,

```

```

    ...content
  };
  return this;
}

toHtml() {
  const templateWrapper = document.createElement('div');
  templateWrapper.innerHTML = this._maybeSanitize(this._config.template);

  for (const [selector, text] of Object.entries(this._config.content)) {
    this._setContent(templateWrapper, text, selector);
  }

  const template = templateWrapper.children[0];

  const extraClass = this._resolvePossibleFunction(this._config.extraClass);

  if (extraClass) {
    template.classList.add(...extraClass.split(' '));
  }

  return template;
} // Private

_typeCheckConfig(config) {
  super._typeCheckConfig(config);

  this._checkContent(config.content);
}

_checkContent(arg) {
  for (const [selector, content] of Object.entries(arg)) {
    super._typeCheckConfig({
      selector,
      entry: content
    }, DefaultContentType);
  }
}

_setContent(template, content, selector) {
  const templateElement = SelectorEngine.findOne(selector, template);

  if (!templateElement) {
    return;
  }

  content = this._resolvePossibleFunction(content);

  if (!content) {
    templateElement.remove();
    return;
  }

  if (isElement$1(content)) {
    this._putElementInTemplate(getElement(content), templateElement);
  }
}

```

```

        return;
    }

    if (this._config.html) {
        templateElement.innerHTML = this._maybeSanitize(content);
        return;
    }

    templateElement.textContent = content;
}

_maybeSanitize(arg) {
    return this._config.sanitize ? sanitizeHtml(arg, this._config.allowList, this._config.sanitizeFn) : arg;
}

_resolvePossibleFunction(arg) {
    return typeof arg === 'function' ? arg(this) : arg;
}

_putElementInTemplate(element, templateElement) {
    if (this._config.html) {
        templateElement.innerHTML = "";
        templateElement.append(element);
        return;
    }

    templateElement.textContent = element.textContent;
}

}

/**
 * -----
 * Bootstrap (v5.2.2): tooltip.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$4 = 'tooltip';
const DISALLOWED_ATTRIBUTES = new Set(['sanitize', 'allowList', 'sanitizeFn']);
const CLASS_NAME_FADE$2 = 'fade';
const CLASS_NAME_MODAL = 'modal';
const CLASS_NAME_SHOW$2 = 'show';
const SELECTOR_TOOLTIP_INNER = '.tooltip-inner';
const SELECTOR_MODAL = `.${CLASS_NAME_MODAL}`;
const EVENT_MODAL_HIDE = 'hide.bs.modal';
const TRIGGER_HOVER = 'hover';
const TRIGGER_FOCUS = 'focus';
const TRIGGER_CLICK = 'click';
const TRIGGER_MANUAL = 'manual';
const EVENT_HIDE$2 = 'hide';
const EVENT_HIDDEN$2 = 'hidden';
const EVENT_SHOW$2 = 'show';
const EVENT_SHOWN$2 = 'shown';

```

```

const EVENT_INSERTED = 'inserted';
const EVENT_CLICK$1 = 'click';
const EVENT_FOCUSIN$1 = 'focusin';
const EVENT_FOCUSOUT$1 = 'focusout';
const EVENT_MOUSEENTER = 'mouseenter';
const EVENT_MOUSELEAVE = 'mouseleave';
const AttachmentMap = {
  AUTO: 'auto',
  TOP: 'top',
  RIGHT: isRTL() ? 'left' : 'right',
  BOTTOM: 'bottom',
  LEFT: isRTL() ? 'right' : 'left'
};
const Default$3 = {
  allowList: DefaultAllowlist,
  animation: true,
  boundary: 'clippingParents',
  container: false,
  customClass: '',
  delay: 0,
  fallbackPlacements: ['top', 'right', 'bottom', 'left'],
  html: false,
  offset: [0, 0],
  placement: 'top',
  popperConfig: null,
  sanitize: true,
  sanitizeFn: null,
  selector: false,
  template: '<div class="tooltip" role="tooltip">' + '<div class="tooltip-arrow"></div>' + '<div
class="tooltip-inner"></div>' + '</div>',
  title: '',
  trigger: 'hover focus'
};
const DefaultType$3 = {
  allowList: 'object',
  animation: 'boolean',
  boundary: '(string|element)',
  container: '(string|element|boolean)',
  customClass: '(string|function)',
  delay: '(number|object)',
  fallbackPlacements: 'array',
  html: 'boolean',
  offset: '(array|string|function)',
  placement: '(string|function)',
  popperConfig: '(null|object|function)',
  sanitize: 'boolean',
  sanitizeFn: '(null|function)',
  selector: '(string|boolean)',
  template: 'string',
  title: '(string|element|function)',
  trigger: 'string'
};
/**
 * Class definition
 */

```

```

class Tooltip extends BaseComponent {

```

```

constructor(element, config) {
  if (typeof Popper === 'undefined') {
    throw new TypeError('Bootstrap\'s tooltips require Popper (https://popper.js.org));
  }

  super(element, config); // Private

  this._isEnabled = true;
  this._timeout = 0;
  this._isHovered = null;
  this._activeTrigger = {};
  this._popper = null;
  this._templateFactory = null;
  this._newContent = null; // Protected

  this.tip = null;

  this._setListeners();

  if (!this._config.selector) {
    this._fixTitle();
  }
} // Getters

static get Default() {
  return Default$3;
}

static get DefaultType() {
  return DefaultType$3;
}

static get NAME() {
  return NAME$4;
} // Public

enable() {
  this._isEnabled = true;
}

disable() {
  this._isEnabled = false;
}

toggleEnabled() {
  this._isEnabled = !this._isEnabled;
}

toggle() {
  if (!this._isEnabled) {
    return;
  }

  this._activeTrigger.click = !this._activeTrigger.click;

```

```

    if (this._isShown()) {
      this._leave();

      return;
    }

    this._enter();
  }

  dispose() {
    clearTimeout(this._timeout);
    EventHandler.off(this._element.closest(SELECTOR_MODAL), EVENT_MODAL_HIDE,
this._hideModalHandler);

    if (this.tip) {
      this.tip.remove();
    }

    if (this._element.getAttribute('data-bs-original-title')) {
      this._element.setAttribute('title', this._element.getAttribute('data-bs-original-title'));
    }

    this._disposePopper();

    super.dispose();
  }

  show() {
    if (this._element.style.display === 'none') {
      throw new Error('Please use show on visible elements');
    }

    if (!(this._isWithContent() && this._isEnabled)) {
      return;
    }

    const showEvent = EventHandler.trigger(this._element,
this.constructor.eventName(EVENT_SHOW$2));
    const shadowRoot = findShadowRoot(this._element);

    const isInTheDom = (shadowRoot ||
this._element.ownerDocument.documentElement).contains(this._element);

    if (showEvent.defaultPrevented || !isInTheDom) {
      return;
    } // todo v6 remove this OR make it optional

    if (this.tip) {
      this.tip.remove();
      this.tip = null;
    }

    const tip = this._getTipElement();

    this._element.setAttribute('aria-describedby', tip.getAttribute('id'));

```

```

const {
  container
} = this._config;

if (!this._element.ownerDocument.documentElement.contains(this.tip)) {
  container.append(tip);
  EventHandler.trigger(this._element, this.constructor.eventName(EVENT_INSERTED));
}

if (this._popper) {
  this._popper.update();
} else {
  this._popper = this._createPopper(tip);
}

tip.classList.add(CLASS_NAME_SHOW$2); // If this is a touch-enabled device we add extra
// empty mouseover listeners to the body's immediate children;
// only needed because of broken event delegation on iOS
// https://www.quirksmode.org/blog/archives/2014/02/mouse_event_bub.html

if ('ontouchstart' in document.documentElement) {
  for (const element of [].concat(...document.body.children)) {
    EventHandler.on(element, 'mouseover', noop);
  }
}

const complete = () => {
  EventHandler.trigger(this._element, this.constructor.eventName(EVENT_SHOWN$2));

  if (this._isHovered === false) {
    this._leave();
  }

  this._isHovered = false;
};

this._queueCallback(complete, this.tip, this._isAnimated());
}

hide() {
  if (!this._isShown()) {
    return;
  }

  const hideEvent = EventHandler.trigger(this._element,
this.constructor.eventName(EVENT_HIDE$2));

  if (hideEvent.defaultPrevented) {
    return;
  }

  const tip = this._getTipElement();

  tip.classList.remove(CLASS_NAME_SHOW$2); // If this is a touch-enabled device we remove the
extra
// empty mouseover listeners we added for iOS support

```



```

if ('ontouchstart' in document.documentElement) {
  for (const element of [].concat(...document.body.children)) {
    EventHandler.off(element, 'mouseover', noop);
  }
}

this._activeTrigger[TRIGGER_CLICK] = false;
this._activeTrigger[TRIGGER_FOCUS] = false;
this._activeTrigger[TRIGGER_HOVER] = false;
this._isHovered = null; // it is a trick to support manual triggering

const complete = () => {
  if (this._isWithActiveTrigger()) {
    return;
  }

  if (!this._isHovered) {
    tip.remove();
  }

  this._element.removeAttribute('aria-describedby');

  EventHandler.trigger(this._element, this.constructor.eventName(EVENT_HIDDEN$2));

  this._disposePopper();
};

this._queueCallback(complete, this.tip, this._isAnimated());
}

update() {
  if (this._popper) {
    this._popper.update();
  }
} // Protected

_isWithContent() {
  return Boolean(this._getTitle());
}

_getTipElement() {
  if (!this.tip) {
    this.tip = this._createTipElement(this._newContent || this._getContentForTemplate());
  }

  return this.tip;
}

_createTipElement(content) {
  const tip = this._getTemplateFactory(content).toHtml(); // todo: remove this check on v6

  if (!tip) {
    return null;
  }

```

tip.classList.remove(CLASS\_NAME\_FADE\$2, CLASS\_NAME\_SHOW\$2); // todo: on v6 the following can be achieved with CSS only

```
tip.classList.add(`bs-${this.constructor.NAME}-auto`);
const tipId = getUID(this.constructor.NAME).toString();
tip.setAttribute('id', tipId);

if (this._isAnimated()) {
  tip.classList.add(CLASS_NAME_FADE$2);
}

return tip;
}

setContent(content) {
  this._newContent = content;

  if (this._isShown()) {
    this._disposePopper();

    this.show();
  }
}

_getTemplateFactory(content) {
  if (this._templateFactory) {
    this._templateFactory.changeContent(content);
  } else {
    this._templateFactory = new TemplateFactory({ ...this._config,
      // the `content` var has to be after `this._config`
      // to override config.content in case of popover
      content,
      extraClass: this._resolvePossibleFunction(this._config.customClass)
    });
  }
}

return this._templateFactory;
}

_getContentForTemplate() {
  return {
    [SELECTOR_TOOLTIP_INNER]: this._getTitle()
  };
}

_getTitle() {
  return this._resolvePossibleFunction(this._config.title) || this._element.getAttribute('data-bs-original-title');
} // Private

_initializeOnDelegatedTarget(event) {
  return this.constructor.getOrCreateInstance(event.delegateTarget, this._getDelegateConfig());
}

_isAnimated() {
  return this._config.animation || this.tip && this.tip.classList.contains(CLASS_NAME_FADE$2);
}
```

```

    }

    _isShown() {
      return this.tip && this.tip.classList.contains(CLASS_NAME_SHOW$2);
    }

    _createPopper(tip) {
      const placement = typeof this._config.placement === 'function' ? this._config.placement.call(this,
tip, this._element) : this._config.placement;
      const attachment = AttachmentMap[placement.toUpperCase()];
      return createPopper(this._element, tip, this._getPopperConfig(attachment));
    }

    _getOffset() {
      const {
        offset
      } = this._config;

      if (typeof offset === 'string') {
        return offset.split(',').map(value => Number.parseInt(value, 10));
      }

      if (typeof offset === 'function') {
        return popperData => offset(popperData, this._element);
      }

      return offset;
    }

    _resolvePossibleFunction(arg) {
      return typeof arg === 'function' ? arg.call(this._element) : arg;
    }

    _getPopperConfig(attachment) {
      const defaultBsPopperConfig = {
        placement: attachment,
        modifiers: [{
          name: 'flip',
          options: {
            fallbackPlacements: this._config.fallbackPlacements
          }
        }, {
          name: 'offset',
          options: {
            offset: this._getOffset()
          }
        }, {
          name: 'preventOverflow',
          options: {
            boundary: this._config.boundary
          }
        }, {
          name: 'arrow',
          options: {
            element: `.${this.constructor.NAME}-arrow`
          }
        }, {

```

```

    name: 'preSetPlacement',
    enabled: true,
    phase: 'beforeMain',
    fn: data => {
      // Pre-set Popper's placement attribute in order to read the arrow sizes properly.
      // Otherwise, Popper mixes up the width and height dimensions since the initial arrow style is
      for top placement
        this._getTipElement().setAttribute('data-popper-placement', data.state.placement);
    }
  }
};
return { ...defaultBsPopperConfig,
  ...(typeof this._config.popperConfig === 'function' ?
this._config.popperConfig(defaultBsPopperConfig) : this._config.popperConfig)
};
}

_setListeners() {
  const triggers = this._config.trigger.split(' ');

  for (const trigger of triggers) {
    if (trigger === 'click') {
      EventHandler.on(this._element, this.constructor.eventName(EVENT_CLICK$1),
this._config.selector, event => {
        const context = this._initializeOnDelegatedTarget(event);

        context.toggle();
      });
    } else if (trigger !== TRIGGER_MANUAL) {
      const eventIn = trigger === TRIGGER_HOVER ?
this.constructor.eventName(EVENT_MOUSEENTER) :
this.constructor.eventName(EVENT_FOCUSIN$1);
      const eventOut = trigger === TRIGGER_HOVER ?
this.constructor.eventName(EVENT_MOUSELEAVE) :
this.constructor.eventName(EVENT_FOCUSOUT$1);
      EventHandler.on(this._element, eventIn, this._config.selector, event => {
        const context = this._initializeOnDelegatedTarget(event);

        context._activeTrigger[event.type === 'focusin' ? TRIGGER_FOCUS : TRIGGER_HOVER] = true;

        context._enter();
      });
      EventHandler.on(this._element, eventOut, this._config.selector, event => {
        const context = this._initializeOnDelegatedTarget(event);

        context._activeTrigger[event.type === 'focusout' ? TRIGGER_FOCUS : TRIGGER_HOVER] =
context._element.contains(event.relatedTarget);

        context._leave();
      });
    }
  }
}

this._hideModalHandler = () => {
  if (this._element) {
    this.hide();
  }
}

```

```

    };

    EventHandler.on(this._element.closest(SELECTOR_MODAL), EVENT_MODAL_HIDE,
this._hideModalHandler);
  }

  _fixTitle() {
    const title = this._element.getAttribute('title');

    if (!title) {
      return;
    }

    if (!this._element.getAttribute('aria-label') && !this._element.textContent.trim()) {
      this._element.setAttribute('aria-label', title);
    }

    this._element.setAttribute('data-bs-original-title', title); // DO NOT USE IT. Is only for backwards
compatibility

    this._element.removeAttribute('title');
  }

  _enter() {
    if (this._isShown() || this._isHovered) {
      this._isHovered = true;
      return;
    }

    this._isHovered = true;

    this._setTimeout(() => {
      if (this._isHovered) {
        this.show();
      }
    }, this._config.delay.show);
  }

  _leave() {
    if (this._isWithActiveTrigger()) {
      return;
    }

    this._isHovered = false;

    this._setTimeout(() => {
      if (!this._isHovered) {
        this.hide();
      }
    }, this._config.delay.hide);
  }

  _setTimeout(handler, timeout) {
    clearTimeout(this._timeout);
    this._timeout = setTimeout(handler, timeout);
  }

```

```

_isWithActiveTrigger() {
  return Object.values(this._activeTrigger).includes(true);
}

_getConfig(config) {
  const dataAttributes = Manipulator.getDataAttributes(this._element);

  for (const dataAttribute of Object.keys(dataAttributes)) {
    if (DISALLOWED_ATTRIBUTES.has(dataAttribute)) {
      delete dataAttributes[dataAttribute];
    }
  }

  config = { ...dataAttributes,
    ...(typeof config === 'object' && config ? config : {})
  };
  config = this._mergeConfigObj(config);
  config = this._configAfterMerge(config);

  this._typeCheckConfig(config);

  return config;
}

_configAfterMerge(config) {
  config.container = config.container === false ? document.body : getElement(config.container);

  if (typeof config.delay === 'number') {
    config.delay = {
      show: config.delay,
      hide: config.delay
    };
  }

  if (typeof config.title === 'number') {
    config.title = config.title.toString();
  }

  if (typeof config.content === 'number') {
    config.content = config.content.toString();
  }

  return config;
}

_getDelegateConfig() {
  const config = {};

  for (const key in this._config) {
    if (this.constructor.Default[key] !== this._config[key]) {
      config[key] = this._config[key];
    }
  }

  config.selector = false;
  config.trigger = 'manual'; // In the future can be replaced with:

```

```

    // const keysWithDifferentValues = Object.entries(this._config).filter(entry =>
this.constructor.Default[entry[0]] !== this._config[entry[0]])
    // `Object.fromEntries(keysWithDifferentValues)`

    return config;
  }

  _disposePopper() {
    if (this._popper) {
      this._popper.destroy();

      this._popper = null;
    }
  } // Static

static jQueryInterface(config) {
  return this.each(function () {
    const data = Tooltip.getOrCreateInstance(this, config);

    if (typeof config !== 'string') {
      return;
    }

    if (typeof data[config] === 'undefined') {
      throw new TypeError(`No method named "${config}"`);
    }

    data[config]();
  });
}

/**
 * jQuery
 */

definejQueryPlugin(Tip);

/**
 * -----
 * Bootstrap (v5.2.2): popover.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$3 = 'popover';
const SELECTOR_TITLE = '.popover-header';
const SELECTOR_CONTENT = '.popover-body';
const Default$2 = { ...Tooltip.Default,
  content: '',
  offset: [0, 8],
  placement: 'right',

```

```

    template: '<div class="popover" role="tooltip">' + '<div class="popover-arrow"></div>' + '<h3
class="popover-header"></h3>' + '<div class="popover-body"></div>' + '</div>',
    trigger: 'click'
  };
  const DefaultType$2 = { ...Tooltip.DefaultType,
    content: '(null|string|element|function)'
  };
  /**
   * Class definition
   */

  class Popover extends Tooltip {
    // Getters
    static get Default() {
      return Default$2;
    }

    static get DefaultType() {
      return DefaultType$2;
    }

    static get NAME() {
      return NAME$3;
    } // Overrides

    _isWithContent() {
      return this._getTitle() || this._getContent();
    } // Private

    _getContentForTemplate() {
      return {
        [SELECTOR_TITLE]: this._getTitle(),
        [SELECTOR_CONTENT]: this._getContent()
      };
    }

    _getContent() {
      return this._resolvePossibleFunction(this._config.content);
    } // Static

    static jQueryInterface(config) {
      return this.each(function () {
        const data = Popover.getInstance(this, config);

        if (typeof config !== 'string') {
          return;
        }

        if (typeof data[config] === 'undefined') {
          throw new TypeError(`No method named "${config}"`);
        }

        data[config]();
      });
    }
  }

```



```

    }

}
/**
 * jQuery
 */

defineJQueryPlugin(Popover);

/**
 * -----
 * Bootstrap (v5.2.2): scrollspy.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$2 = 'scrollspy';
const DATA_KEY$2 = 'bs.scrollspy';
const EVENT_KEY$2 = `.${DATA_KEY$2}`;
const DATA_API_KEY = '.data-api';
const EVENT_ACTIVATE = `activate${EVENT_KEY$2}`;
const EVENT_CLICK = `click${EVENT_KEY$2}`;
const EVENT_LOAD_DATA_API$1 = `load${EVENT_KEY$2}${DATA_API_KEY}`;
const CLASS_NAME_DROPDOWN_ITEM = 'dropdown-item';
const CLASS_NAME_ACTIVE$1 = 'active';
const SELECTOR_DATA_SPY = '[data-bs-spy="scroll"]';
const SELECTOR_TARGET_LINKS = '[href]';
const SELECTOR_NAV_LIST_GROUP = '.nav, .list-group';
const SELECTOR_NAV_LINKS = '.nav-link';
const SELECTOR_NAV_ITEMS = '.nav-item';
const SELECTOR_LIST_ITEMS = '.list-group-item';
const SELECTOR_LINK_ITEMS = `${SELECTOR_NAV_LINKS}, ${SELECTOR_NAV_ITEMS} >
${SELECTOR_NAV_LINKS}, ${SELECTOR_LIST_ITEMS}`;
const SELECTOR_DROPDOWN = '.dropdown';
const SELECTOR_DROPDOWN_TOGGLE$1 = '.dropdown-toggle';
const Default$1 = {
  offset: null,
  // TODO: v6 @deprecated, keep it for backwards compatibility reasons
  rootMargin: '0px 0px -25%',
  smoothScroll: false,
  target: null,
  threshold: [0.1, 0.5, 1]
};
const DefaultType$1 = {
  offset: '(number|null)',
  // TODO v6 @deprecated, keep it for backwards compatibility reasons
  rootMargin: 'string',
  smoothScroll: 'boolean',
  target: 'element',
  threshold: 'array'
};
/**
 * Class definition

```

```
*/
```

```
class ScrollSpy extends BaseComponent {
  constructor(element, config) {
    super(element, config); // this._element is the observablesContainer and config.target the menu
links wrapper

    this._targetLinks = new Map();
    this._observableSections = new Map();
    this._rootElement = getComputedStyle(this._element).overflowY === 'visible' ? null : this._element;
    this._activeTarget = null;
    this._observer = null;
    this._previousScrollData = {
      visibleEntryTop: 0,
      parentScrollTop: 0
    };
    this.refresh(); // initialize
  } // Getters

  static get Default() {
    return Default$1;
  }

  static get DefaultType() {
    return DefaultType$1;
  }

  static get NAME() {
    return NAME$2;
  } // Public

  refresh() {
    this._initializeTargetsAndObservables();

    this._maybeEnableSmoothScroll();

    if (this._observer) {
      this._observer.disconnect();
    } else {
      this._observer = this._getNewObserver();
    }

    for (const section of this._observableSections.values()) {
      this._observer.observe(section);
    }
  }

  dispose() {
    this._observer.disconnect();

    super.dispose();
  } // Private

  _configAfterMerge(config) {
```

```
// TODO: on v6 target should be given explicitly & remove the {target: 'ss-target'} case
config.target = getElement(config.target) || document.body; // TODO: v6 Only for backwards
compatibility reasons. Use rootMargin only
```

```
config.rootMargin = config.offset ? `${config.offset}px 0px -30%` : config.rootMargin;

if (typeof config.threshold === 'string') {
  config.threshold = config.threshold.split(',').map(value => Number.parseFloat(value));
}

return config;
}
```

```
_maybeEnableSmoothScroll() {
  if (!this._config.smoothScroll) {
    return;
  } // unregister any previous listeners
```

```
EventHandler.off(this._config.target, EVENT_CLICK);
EventHandler.on(this._config.target, EVENT_CLICK, SELECTOR_TARGET_LINKS, event => {
  const observableSection = this._observableSections.get(event.target.hash);
```

```
  if (observableSection) {
    event.preventDefault();
    const root = this._rootElement || window;
    const height = observableSection.offsetTop - this._element.offsetTop;
```

```
    if (root.scrollTo) {
      root.scrollTo({
        top: height,
        behavior: 'smooth'
      });
      return;
    } // Chrome 60 doesn't support `scrollTo`
```

```
    root.scrollTop = height;
  }
});
}
```

```
_getNewObserver() {
  const options = {
    root: this._rootElement,
    threshold: this._config.threshold,
    rootMargin: this._config.rootMargin
  };
  return new IntersectionObserver(entries => this._observerCallback(entries), options);
} // The logic of selection
```

```
_observerCallback(entries) {
  const targetElement = entry => this._targetLinks.get(`#${entry.target.id}`);
```

```
  const activate = entry => {
    this._previousScrollData.visibleEntryTop = entry.target.offsetTop;
```

```

    this._process(targetElement(entry));
};

const parentScrollTop = (this._rootElement || document.documentElement).scrollTop;
const userScrollsDown = parentScrollTop >= this._previousScrollData.parentScrollTop;
this._previousScrollData.parentScrollTop = parentScrollTop;

for (const entry of entries) {
    if (!entry.isIntersecting) {
        this._activeTarget = null;

        this._clearActiveClass(targetElement(entry));

        continue;
    }

    const entryIsLowerThanPrevious = entry.target.offsetTop >=
this._previousScrollData.visibleEntryTop; // if we are scrolling down, pick the bigger offsetTop

    if (userScrollsDown && entryIsLowerThanPrevious) {
        activate(entry); // if parent isn't scrolled, let's keep the first visible item, breaking the iteration

        if (!parentScrollTop) {
            return;
        }

        continue;
    } // if we are scrolling up, pick the smallest offsetTop

    if (!userScrollsDown && !entryIsLowerThanPrevious) {
        activate(entry);
    }
}

_initializeTargetsAndObservables() {
    this._targetLinks = new Map();
    this._observableSections = new Map();
    const targetLinks = SelectorEngine.find(SELECTOR_TARGET_LINKS, this._config.target);

    for (const anchor of targetLinks) {
        // ensure that the anchor has an id and is not disabled
        if (!anchor.hash || isDisabled(anchor)) {
            continue;
        }

        const observableSection = SelectorEngine.findOne(anchor.hash, this._element); // ensure that
the observableSection exists & is visible

        if (isVisible(observableSection)) {
            this._targetLinks.set(anchor.hash, anchor);

            this._observableSections.set(anchor.hash, observableSection);
        }
    }
}

```

```

}

_process(target) {
  if (this._activeTarget === target) {
    return;
  }

  this._clearActiveClass(this._config.target);

  this._activeTarget = target;
  target.classList.add(CLASS_NAME_ACTIVE$1);

  this._activateParents(target);

  EventHandler.trigger(this._element, EVENT_ACTIVATE, {
    relatedTarget: target
  });
}

_activateParents(target) {
  // Activate dropdown parents
  if (target.classList.contains(CLASS_NAME_DROPDOWN_ITEM)) {
    SelectorEngine.findOne(SELECTOR_DROPDOWN_TOGGLE$1,
    target.closest(SELECTOR_DROPDOWN)).classList.add(CLASS_NAME_ACTIVE$1);
    return;
  }

  for (const listGroup of SelectorEngine.parents(target, SELECTOR_NAV_LIST_GROUP)) {
    // Set triggered links parents as active
    // With both <ul> and <nav> markup a parent is the previous sibling of any nav ancestor
    for (const item of SelectorEngine.prev(listGroup, SELECTOR_LINK_ITEMS)) {
      item.classList.add(CLASS_NAME_ACTIVE$1);
    }
  }
}

_clearActiveClass(parent) {
  parent.classList.remove(CLASS_NAME_ACTIVE$1);
  const activeNodes =
  SelectorEngine.find(`${SELECTOR_TARGET_LINKS}.${CLASS_NAME_ACTIVE$1}`, parent);

  for (const node of activeNodes) {
    node.classList.remove(CLASS_NAME_ACTIVE$1);
  }
} // Static

static jQueryInterface(config) {
  return this.each(function () {
    const data = ScrollSpy.getInstance(this, config);

    if (typeof config !== 'string') {
      return;
    }

    if (data[config] === undefined || config.startsWith('_') || config === 'constructor') {
      throw new TypeError(`No method named "${config}"`);
    }
  });
}

```

```

    }

    data[config]();
  });
}

}
/**
 * Data API implementation
 */

EventHandler.on(window, EVENT_LOAD_DATA_API$1, () => {
  for (const spy of SelectorEngine.find(SELECTOR_DATA_SPY)) {
    ScrollSpy.getOrCreateInstance(spy);
  }
});
/**
 * jQuery
 */

defineJQueryPlugin(ScrollSpy);

/**
 * -----
 * Bootstrap (v5.2.2): tab.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME$1 = 'tab';
const DATA_KEY$1 = 'bs.tab';
const EVENT_KEY$1 = `.${DATA_KEY$1}`;
const EVENT_HIDE$1 = `hide${EVENT_KEY$1}`;
const EVENT_HIDDEN$1 = `hidden${EVENT_KEY$1}`;
const EVENT_SHOW$1 = `show${EVENT_KEY$1}`;
const EVENT_SHOWN$1 = `shown${EVENT_KEY$1}`;
const EVENT_CLICK_DATA_API = `click${EVENT_KEY$1}`;
const EVENT_KEYDOWN = `keydown${EVENT_KEY$1}`;
const EVENT_LOAD_DATA_API = `load${EVENT_KEY$1}`;
const ARROW_LEFT_KEY = 'ArrowLeft';
const ARROW_RIGHT_KEY = 'ArrowRight';
const ARROW_UP_KEY = 'ArrowUp';
const ARROW_DOWN_KEY = 'ArrowDown';
const CLASS_NAME_ACTIVE = 'active';
const CLASS_NAME_FADE$1 = 'fade';
const CLASS_NAME_SHOW$1 = 'show';
const CLASS_DROPDOWN = 'dropdown';
const SELECTOR_DROPDOWN_TOGGLE = '.dropdown-toggle';
const SELECTOR_DROPDOWN_MENU = '.dropdown-menu';
const NOT_SELECTOR_DROPDOWN_TOGGLE = ':not(.dropdown-toggle)';
const SELECTOR_TAB_PANEL = '.list-group, .nav, [role="tablist"]';
const SELECTOR_OUTER = '.nav-item, .list-group-item';

```

```

const SELECTOR_INNER = `.nav-link${NOT_SELECTOR_DROPDOWN_TOGGLE}, .list-group-
item${NOT_SELECTOR_DROPDOWN_TOGGLE}, [role="tab"]${NOT_SELECTOR_DROPDOWN_TOGGLE}`;
const SELECTOR_DATA_TOGGLE = '[data-bs-toggle="tab"], [data-bs-toggle="pill"], [data-bs-
toggle="list"]'; // todo:v6: could be only `tab`

```

```

const SELECTOR_INNER_ELEM = `${SELECTOR_INNER}, ${SELECTOR_DATA_TOGGLE}`;
const SELECTOR_DATA_TOGGLE_ACTIVE = `.${CLASS_NAME_ACTIVE}[data-bs-
toggle="tab"], .${CLASS_NAME_ACTIVE}[data-bs-toggle="pill"], .${CLASS_NAME_ACTIVE}[data-bs-
toggle="list"]`;

```

```

/**

```

```

 * Class definition

```

```

 */

```

```

class Tab extends BaseComponent {

```

```

  constructor(element) {

```

```

    super(element);

```

```

    this._parent = this._element.closest(SELECTOR_TAB_PANEL);

```

```

    if (!this._parent) {

```

```

      return; // todo: should Throw exception on v6

```

```

      // throw new TypeError(`${element.outerHTML} has not a valid parent

```

```

      ${SELECTOR_INNER_ELEM}`)

```

```

    } // Set up initial aria attributes

```

```

    this._setInitialAttributes(this._parent, this._getChildren());

```

```

    EventHandler.on(this._element, EVENT_KEYDOWN, event => this._keydown(event));

```

```

  } // Getters

```

```

  static get NAME() {

```

```

    return NAME$1;

```

```

  } // Public

```

```

  show() {

```

```

    // Shows this elem and deactivate the active sibling if exists

```

```

    const innerElem = this._element;

```

```

    if (this._elementIsActive(innerElem)) {

```

```

      return;

```

```

    } // Search for active tab on same parent to deactivate it

```

```

    const active = this._getActiveElem();

```

```

    const hideEvent = active ? EventHandler.trigger(active, EVENT_HIDE$1, {
      relatedTarget: innerElem
    }) : null;

```

```

    const showEvent = EventHandler.trigger(innerElem, EVENT_SHOW$1, {
      relatedTarget: active
    });

```

```

    if (showEvent.defaultPrevented || hideEvent && hideEvent.defaultPrevented) {
      return;
    }

```

```

    this._deactivate(active, innerElem);

    this._activate(innerElem, active);
} // Private

_activate(element, relatedElem) {
    if (!element) {
        return;
    }

    element.classList.add(CLASS_NAME_ACTIVE);

    this._activate(getElementFromSelector(element)); // Search and activate/show the proper section

    const complete = () => {
        if (element.getAttribute('role') !== 'tab') {
            element.classList.add(CLASS_NAME_SHOW$1);
            return;
        }

        element.removeAttribute('tabindex');
        element.setAttribute('aria-selected', true);

        this._toggleDropDown(element, true);

        EventHandler.trigger(element, EVENT_SHOWN$1, {
            relatedTarget: relatedElem
        });
    };

    this._queueCallback(complete, element, element.classList.contains(CLASS_NAME_FADE$1));
}

_deactivate(element, relatedElem) {
    if (!element) {
        return;
    }

    element.classList.remove(CLASS_NAME_ACTIVE);
    element.blur();

    this._deactivate(getElementFromSelector(element)); // Search and deactivate the shown section
    too

    const complete = () => {
        if (element.getAttribute('role') !== 'tab') {
            element.classList.remove(CLASS_NAME_SHOW$1);
            return;
        }

        element.setAttribute('aria-selected', false);
        element.setAttribute('tabindex', '-1');
    };

```



```

    this._toggleDropDown(element, false);

    EventHandler.trigger(element, EVENT_HIDDEN$1, {
      relatedTarget: relatedElem
    });
  };

  this._queueCallback(complete, element, element.classList.contains(CLASS_NAME_FADE$1));
}

_keydown(event) {
  if (![ARROW_LEFT_KEY, ARROW_RIGHT_KEY, ARROW_UP_KEY,
  ARROW_DOWN_KEY].includes(event.key)) {
    return;
  }

  event.stopPropagation(); // stopPropagation/preventDefault both added to support up/down keys
  without scrolling the page

  event.preventDefault();
  const isNext = [ARROW_RIGHT_KEY, ARROW_DOWN_KEY].includes(event.key);
  const nextActiveElement = getNextActiveElement(this._getChildren().filter(element
=> !isDisabled(element)), event.target, isNext, true);

  if (nextActiveElement) {
    nextActiveElement.focus({
      preventScroll: true
    });
    Tab.getOrCreateInstance(nextActiveElement).show();
  }
}

_getChildren() {
  // collection of inner elements
  return SelectorEngine.find(SELECTOR_INNER_ELEM, this._parent);
}

_getActiveElem() {
  return this._getChildren().find(child => this._elemIsActive(child)) || null;
}

_setInitialAttributes(parent, children) {
  this._setAttributesIfNotExists(parent, 'role', 'tablist');

  for (const child of children) {
    this._setInitialAttributesOnChild(child);
  }
}

_setInitialAttributesOnChild(child) {
  child = this._getInnerElement(child);

  const isActive = this._elemIsActive(child);

  const outerElem = this._getOuterElement(child);

  child.setAttribute('aria-selected', isActive);

```

```

    if (outerElem !== child) {
      this._setAttributeIfNotExists(outerElem, 'role', 'presentation');
    }

    if (!isActive) {
      child.setAttribute('tabindex', '-1');
    }

    this._setAttributeIfNotExists(child, 'role', 'tab'); // set attributes to the related panel too

    this._setInitialAttributesOnTargetPanel(child);
  }

  _setInitialAttributesOnTargetPanel(child) {
    const target = getElementFromSelector(child);

    if (!target) {
      return;
    }

    this._setAttributeIfNotExists(target, 'role', 'tabpanel');

    if (child.id) {
      this._setAttributeIfNotExists(target, 'aria-labelledby', `${child.id}`);
    }
  }

  _toggleDropDown(element, open) {
    const outerElem = this._getOuterElement(element);

    if (!outerElem.classList.contains(CLASS_DROPDOWN)) {
      return;
    }

    const toggle = (selector, className) => {
      const element = SelectorEngine.findOne(selector, outerElem);

      if (element) {
        element.classList.toggle(className, open);
      }
    };

    toggle(SELECTOR_DROPDOWN_TOGGLE, CLASS_NAME_ACTIVE);
    toggle(SELECTOR_DROPDOWN_MENU, CLASS_NAME_SHOW$1);
    outerElem.setAttribute('aria-expanded', open);
  }

  _setAttributeIfNotExists(element, attribute, value) {
    if (!element.hasAttribute(attribute)) {
      element.setAttribute(attribute, value);
    }
  }

  _elemsActive(elem) {
    return elem.classList.contains(CLASS_NAME_ACTIVE);
  }

```

```

    } // Try to get the inner element (usually the .nav-link)

    _getInnerElement(elem) {
        return elem.matches(SELECTOR_INNER_ELEM) ? elem :
SelectorEngine.findOne(SELECTOR_INNER_ELEM, elem);
    } // Try to get the outer element (usually the .nav-item)

    _getOuterElement(elem) {
        return elem.closest(SELECTOR_OUTER) || elem;
    } // Static

static jQueryInterface(config) {
    return this.each(function () {
        const data = Tab.getOrCreateInstance(this);

        if (typeof config !== 'string') {
            return;
        }

        if (data[config] === undefined || config.startsWith('_') || config === 'constructor') {
            throw new TypeError(`No method named "${config}"`);
        }

        data[config]();
    });
}
/**
 * Data API implementation
 */

EventHandler.on(document, EVENT_CLICK_DATA_API, SELECTOR_DATA_TOGGLE, function (event) {
    if (['A', 'AREA'].includes(this.tagName)) {
        event.preventDefault();
    }

    if (isDisabled(this)) {
        return;
    }

    Tab.getOrCreateInstance(this).show();
});
/**
 * Initialize on focus
 */

EventHandler.on(window, EVENT_LOAD_DATA_API, () => {
    for (const element of SelectorEngine.find(SELECTOR_DATA_TOGGLE_ACTIVE)) {
        Tab.getOrCreateInstance(element);
    }
});
/**

```

```

* jQuery
*/

defineJQueryPlugin(Tab);

/**
 * -----
 * Bootstrap (v5.2.2): toast.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 * -----
 */
/**
 * Constants
 */

const NAME = 'toast';
const DATA_KEY = 'bs.toast';
const EVENT_KEY = `.${DATA_KEY}`;
const EVENT_MOUSEOVER = `mouseover${EVENT_KEY}`;
const EVENT_MOUSEOUT = `mouseout${EVENT_KEY}`;
const EVENT_FOCUSIN = `focusin${EVENT_KEY}`;
const EVENT_FOCUSOUT = `focusout${EVENT_KEY}`;
const EVENT_HIDE = `hide${EVENT_KEY}`;
const EVENT_HIDDEN = `hidden${EVENT_KEY}`;
const EVENT_SHOW = `show${EVENT_KEY}`;
const EVENT_SHOWN = `shown${EVENT_KEY}`;
const CLASS_NAME_FADE = 'fade';
const CLASS_NAME_HIDE = 'hide'; // @deprecated - kept here only for backwards compatibility

const CLASS_NAME_SHOW = 'show';
const CLASS_NAME_SHOWING = 'showing';
const DefaultType = {
  animation: 'boolean',
  autohide: 'boolean',
  delay: 'number'
};
const Default = {
  animation: true,
  autohide: true,
  delay: 5000
};
/**
 * Class definition
 */

class Toast extends BaseComponent {
  constructor(element, config) {
    super(element, config);
    this._timeout = null;
    this._hasMouseInteraction = false;
    this._hasKeyboardInteraction = false;

    this._setListeners();
  } // Getters

  static get Default() {

```

```

    return Default;
}

static get DefaultType() {
    return DefaultType;
}

static get NAME() {
    return NAME;
} // Public

show() {
    const showEvent = EventHandler.trigger(this._element, EVENT_SHOW);

    if (showEvent.defaultPrevented) {
        return;
    }

    this._clearTimeout();

    if (this._config.animation) {
        this._element.classList.add(CLASS_NAME_FADE);
    }

    const complete = () => {
        this._element.classList.remove(CLASS_NAME_SHOWING);

        EventHandler.trigger(this._element, EVENT_SHOWN);

        this._maybeScheduleHide();
    };

    this._element.classList.remove(CLASS_NAME_HIDE); // @deprecated

    reflow(this._element);

    this._element.classList.add(CLASS_NAME_SHOW, CLASS_NAME_SHOWING);

    this._queueCallback(complete, this._element, this._config.animation);
}

hide() {
    if (!this.isShown()) {
        return;
    }

    const hideEvent = EventHandler.trigger(this._element, EVENT_HIDE);

    if (hideEvent.defaultPrevented) {
        return;
    }

    const complete = () => {
        this._element.classList.add(CLASS_NAME_HIDE); // @deprecated
    };

```

```

    this._element.classList.remove(CLASS_NAME_SHOWING, CLASS_NAME_SHOW);

    EventHandler.trigger(this._element, EVENT_HIDDEN);
};

this._element.classList.add(CLASS_NAME_SHOWING);

this._queueCallback(complete, this._element, this._config.animation);
}

dispose() {
    this._clearTimeout();

    if (this.isShown()) {
        this._element.classList.remove(CLASS_NAME_SHOW);
    }

    super.dispose();
}

isShown() {
    return this._element.classList.contains(CLASS_NAME_SHOW);
} // Private

_maybeScheduleHide() {
    if (!this._config.autohide) {
        return;
    }

    if (this._hasMouseInteraction || this._hasKeyboardInteraction) {
        return;
    }

    this._timeout = setTimeout(() => {
        this.hide();
    }, this._config.delay);
}

_onInteraction(event, isInteracting) {
    switch (event.type) {
        case 'mouseover':
        case 'mouseout':
            {
                this._hasMouseInteraction = isInteracting;
                break;
            }

        case 'focusin':
        case 'focusout':
            {
                this._hasKeyboardInteraction = isInteracting;
                break;
            }
    }
}

```

```

    if (isInteracting) {
        this._clearTimeout();

        return;
    }

    const nextElement = event.relatedTarget;

    if (this._element === nextElement || this._element.contains(nextElement)) {
        return;
    }

    this._maybeScheduleHide();
}

_setListeners() {
    EventHandler.on(this._element, EVENT_MOUSEOVER, event => this._onInteraction(event, true));
    EventHandler.on(this._element, EVENT_MOUSEOUT, event => this._onInteraction(event, false));
    EventHandler.on(this._element, EVENT_FOCUSIN, event => this._onInteraction(event, true));
    EventHandler.on(this._element, EVENT_FOCUSOUT, event => this._onInteraction(event, false));
}

_clearTimeout() {
    clearTimeout(this._timeout);
    this._timeout = null;
} // Static

static jQueryInterface(config) {
    return this.each(function () {
        const data = Toast.getOrCreateInstance(this, config);

        if (typeof config === 'string') {
            if (typeof data[config] === 'undefined') {
                throw new TypeError(`No method named "${config}"`);
            }

            data[config](this);
        }
    });
}

/**
 * Data API implementation
 */

enableDismissTrigger(Toast);

/**
 * jQuery
 */

defineJQueryPlugin(Toast);

/**
 * -----

```

```

* Bootstrap (v5.2.2): index.umd.js
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
* -----
*/
const index_umd = {
  Alert,
  Button,
  Carousel,
  Collapse,
  Dropdown,
  Modal,
  Offcanvas,
  Popover,
  ScrollSpy,
  Tab,
  Toast,
  Tooltip
};

return index_umd;

});
//# sourceMappingURL=bootstrap.bundle.js.map

js/bootstrap-js/bootstrap.bundle.js.map

```

### **js/retonoDados/retornoDadosAvaClinic.js**

```

let imcRetornoSexo = document.querySelector('#imc-retorno-sexo');
let imcRetornoIdade = document.querySelector('#imc-retorno-idade');

let circAbdomRetornoSexo = document.querySelector('#circ-abdom-retorno-sexo');
let cintQuadrRetornoSexo = document.querySelector('#cint-quadr-retorno-sexo');

let etapa2RetornoIdade = document.querySelector('#etapa2-retorno-idade');
let etapa2RetornoSexo = document.querySelector('#etapa2-retorno-sexo');
let etapa2ColestTotal = document.querySelector('#etapa2-colest-total');
let etapa2Hdl = document.querySelector('#etapa2-hdl');

let checkMasc = document.querySelector('#br-sex-masc');
let checkIdade = document.querySelector('#br-idade');

function retornaValoresHemograma() {
  etapa2ColestTotal.value = localStorage.getItem('hemogColes');
  etapa2Hdl.value = localStorage.getItem('hemogHdl');
}

returnaValoresHemograma();

/*Função para marcar checkeds, anteriormente informados*/
function verificaCheckeds() {
  let dataAtual = new Date();
  const anoAtual = dataAtual.getFullYear();
  const mesAtual = dataAtual.getMonth() + 1;
  const diaAtual = dataAtual.getDate();

```



```

let dataInformada = 0;
let idade = 0;
if ((localStorage.getItem('userCadData')) == null) {
    dataInformada = '';
    /*Salva idade em localStorage*/
    localStorage.setItem('userCadIdade', dataInformada);

} else {
    dataInformada = localStorage.getItem('userCadData');
    const dataNascimento = new Date(dataInformada);
    const anoNascimento = dataNascimento.getFullYear();
    const mesNascimento = dataNascimento.getMonth() + 1;
    const diaNascimento = dataNascimento.getDate();
    idade = anoAtual - anoNascimento;

    if (mesAtual < mesNascimento || (mesAtual === mesNascimento && diaAtual < diaNascimento))
    {
        idade--;
    }
    /*Salva idade em localStorage*/
    localStorage.setItem('userCadIdade', idade);
}

console.log('Data hj: ', dataAtual, dataInformada, (idade));

if (localStorage.getItem('userCadSexo') === 'Masculino') {
    checkMasc.checked = true;

} else if (localStorage.getItem('userCadSexo') === 'Feminino') {
    checkMasc.checked = false;

}

if (localStorage.getItem('userCadIdade') > 65) {
    checkIdade.checked = true;

} else {
    checkIdade.checked = false;

}

}
verificaCheckeds();

function retornaSexoIdade() {
    imcRetornoIdade.innerHTML = localStorage.getItem('userCadIdade');
    imcRetornoSexo.innerHTML = localStorage.getItem('userCadSexo');

    circAbdomRetornoSexo.innerHTML = localStorage.getItem('userCadSexo');
    cintQuadrRetornoSexo.innerHTML = localStorage.getItem('userCadSexo');

    etapa2RetornoIdade.innerHTML = localStorage.getItem('userCadIdade');
    etapa2RetornoSexo.innerHTML = localStorage.getItem('userCadSexo');

    console.log(localStorage.getItem('userCadSexo'), localStorage.getItem('userCadIdade'));

}
retornaSexoIdade();

```

```

/*Função que preenche input anteriormente informados*/
function retornaDadosInputs() {
  let pas = document.querySelector('#pas');
  let pad = document.querySelector('#pad');

  let peso = document.querySelector('#imc-peso');
  let altura = document.querySelector('#imc-altura');

  let circAbdomCm = document.querySelector('#circ-abdom-cm');

  let cintQuadrAbdomCm = document.querySelector('#cint-quadr-abdom-cm');
  let cintQuadrQuadrCm = document.querySelector('#cint-quadr-quadr-cm');

  let tornBraqPasTorn = document.querySelector('#torn-braq-pas-torn');
  let tornBraqPasBra = document.querySelector('#torn-braq-pas-bra');

  let etapa2ColestTotal = document.querySelector('#etapa2-colest-total');
  let etapa2Fumante = document.querySelector('#etapa2-fumante');
  let etapa2Hdl = document.querySelector('#etapa2-hdl');
  let etapa2PaSistol = document.querySelector('#etapa2-pa-sistol');
  let etapa2Tratada = document.querySelector('#etapa2-tratada');

  pas.value = localStorage.getItem('pas');
  pad.value = localStorage.getItem('pad');

  peso.value = localStorage.getItem('peso');
  altura.value = localStorage.getItem('altura');

  circAbdomCm.value = localStorage.getItem('circAbdomCm');

  cintQuadrAbdomCm.value = localStorage.getItem('cintQuadrAbdomCm');
  cintQuadrQuadrCm.value = localStorage.getItem('cintQuadrQuadrCm');

  tornBraqPasTorn.value = localStorage.getItem('tornBraqPasTorn');
  tornBraqPasBra.value = localStorage.getItem('pas');

  etapa2ColestTotal.value = localStorage.getItem('hemogColes');
  etapa2Hdl.value = localStorage.getItem('hemogHdl');
  etapa2PaSistol.value = localStorage.getItem('pas');

  if ((localStorage.getItem('etapa2Tratada') == null) || (localStorage.getItem('etapa2Tratada') == 'Não
tratada')) {
    console.log('false', localStorage.getItem('etapa2Tratada'));
    etapa2Tratada.checked = false;
  } else {
    console.log('true', localStorage.getItem('etapa2Tratada'));
    etapa2Tratada.checked = true;
  }
  if ((localStorage.getItem('brTabag') == null) || (localStorage.getItem('brTabag') == 'Não fumante')) {
    console.log('false', localStorage.getItem('brTabag'));
    etapa2Fumante.checked = false;
  } else {
    console.log('true', localStorage.getItem('brTabag'));
  }
}

```

```

        etapa2Fumante.checked = true;

    }

}

retornaDadosInputs();

/*Função que preenche input anteriormente informados*/
function retornaDadosCheckbox() {
    /*Variaveis Risco Baixo*/
    let brIdade = document.querySelector('#br-idade');
    let brSexMasc = document.querySelector('#br-sex-masc');
    let brTabag = document.querySelector('#br-tabag');
    let brHipert = document.querySelector('#br-hipert');
    let brObs = document.querySelector('#br-obs');
    let brSedent = document.querySelector('#br-sedent');
    let brHistoFam = document.querySelector('#br-histo-fam');
    let brHistoDoenca = document.querySelector('#br-histo-doenca');
    let brManifest = document.querySelector('#br-manifest');
    let brDiagnDislip = document.querySelector('#br-diagn-dislip');
    let brDiagnPolic = document.querySelector('#br-diagn-polic');

    /*Variaveis Risco Alto */
    let arAcidVasc = document.querySelector('#ar-acid-vasc');
    let arInfartAgud = document.querySelector('#ar-infart-agud');
    let arLesPerif = document.querySelector('#ar-les-perif');
    let arAtaqIsqu = document.querySelector('#ar-ataq-isqu');
    let arHiperVentr = document.querySelector('#ar-hiper-ventr');
    let arNefrop = document.querySelector('#ar-nefrop');
    let arRetin = document.querySelector('#ar-retin');
    let arAneur = document.querySelector('#ar-aneur');
    let arEsten = document.querySelector('#ar-esten');
    let arDiabet = document.querySelector('#ar-diabet');
    let arDoenVasc = document.querySelector('#ar-doen-vasc');
    let arInsuf = document.querySelector('#ar-insuf');
    let arAngina = document.querySelector('#ar-angina');
    let arDoencRenal = document.querySelector('#ar-doenc-renal');

    /*Salva Baixo Risco em LocalStorage */
    localStorage.setItem('arDoenVasc', arDoenVasc);
    localStorage.setItem('arInsuf', arInsuf);
    localStorage.setItem('arAngina', arAngina);
    localStorage.setItem('arDoencRenal', arDoencRenal);
    */

    if ((localStorage.getItem('brTabag') == null) || (localStorage.getItem('brTabag') == 'Não fumante')) {
        brTabag.checked = false;
    } else {
        brTabag.checked = true;
    }

    if ((localStorage.getItem('brHipert') == null) || (localStorage.getItem('brHipert') == 'vazio')) {
        brHipert.checked = false;
    } else {
        brHipert.checked = true;
    }
}

```

```

}
if ((localStorage.getItem('brObs') == null) || (localStorage.getItem('brObs') == 'vazio')) {
    brObs.checked = false;

} else {
    brObs.checked = true;

}
if ((localStorage.getItem('brSedent') == null) || (localStorage.getItem('brSedent') == 'vazio')) {
    brSedent.checked = false;

} else {
    brSedent.checked = true;

}
if ((localStorage.getItem('brHistoFam') == null) || (localStorage.getItem('brHistoFam') == 'vazio')) {
    brHistoFam.checked = false;

} else {
    brHistoFam.checked = true;

}
if ((localStorage.getItem('brHistoDoenca') == null) || (localStorage.getItem('brHistoDoenca') ==
'vazio')) {
    brHistoDoenca.checked = false;

} else {
    brHistoDoenca.checked = true;

}
if ((localStorage.getItem('brManifest') == null) || (localStorage.getItem('brManifest') == 'vazio')) {
    brManifest.checked = false;

} else {
    brManifest.checked = true;

}
if ((localStorage.getItem('brDiagnDislip') == null) || (localStorage.getItem('brDiagnDislip') == 'vazio'))
{
    brDiagnDislip.checked = false;

} else {
    brDiagnDislip.checked = true;

}
if ((localStorage.getItem('brDiagnPolic') == null) || (localStorage.getItem('brDiagnPolic') == 'vazio'))
{
    brDiagnPolic.checked = false;

} else {
    brDiagnPolic.checked = true;

}
if ((localStorage.getItem('arAcidVasc') == null) || (localStorage.getItem('arAcidVasc') == 'vazio')) {
    arAcidVasc.checked = false;

```

```

    } else {
        arAcidVasc.checked = true;

    }
    if ((localStorage.getItem('arInfartAgud') == null) || (localStorage.getItem('arInfartAgud') == 'vazio'))
    {
        arInfartAgud.checked = false;

    } else {
        arInfartAgud.checked = true;

    }
    if ((localStorage.getItem('arLesPerif') == null) || (localStorage.getItem('arLesPerif') == 'vazio')) {
        arLesPerif.checked = false;

    } else {
        arLesPerif.checked = true;

    }
    if ((localStorage.getItem('arAtaqlsqu') == null) || (localStorage.getItem('arAtaqlsqu') == 'vazio')) {
        arAtaqlsqu.checked = false;

    } else {
        arAtaqlsqu.checked = true;

    }
    if ((localStorage.getItem('arHiperVentr') == null) || (localStorage.getItem('arHiperVentr') == 'vazio'))
    {
        arHiperVentr.checked = false;

    } else {
        arHiperVentr.checked = true;

    }
    if ((localStorage.getItem('arNefrop') == null) || (localStorage.getItem('arNefrop') == 'vazio')) {
        arNefrop.checked = false;

    } else {
        arNefrop.checked = true;

    }
    if ((localStorage.getItem('arRetin') == null) || (localStorage.getItem('arRetin') == 'vazio')) {
        arRetin.checked = false;

    } else {
        arRetin.checked = true;

    }
    if ((localStorage.getItem('arAneur') == null) || (localStorage.getItem('arAneur') == 'vazio')) {
        arAneur.checked = false;

    } else {
        arAneur.checked = true;

    }
    if ((localStorage.getItem('arEsten') == null) || (localStorage.getItem('arEsten') == 'vazio')) {
        arEsten.checked = false;

```

```

    } else {
        arEsten.checked = true;
    }
    if ((localStorage.getItem('arDiabet') == null) || (localStorage.getItem('arDiabet') == 'vazio')) {
        arDiabet.checked = false;
    } else {
        arDiabet.checked = true;
    }
    if ((localStorage.getItem('arDoenVasc') == null) || (localStorage.getItem('arDoenVasc') == 'vazio')) {
        arDoenVasc.checked = false;
    } else {
        arDoenVasc.checked = true;
    }
    if ((localStorage.getItem('arInsuf') == null) || (localStorage.getItem('arInsuf') == 'vazio')) {
        arInsuf.checked = false;
    } else {
        arInsuf.checked = true;
    }
    if ((localStorage.getItem('arAngina') == null) || (localStorage.getItem('arAngina') == 'vazio')) {
        arAngina.checked = false;
    } else {
        arAngina.checked = false;
    }
    if ((localStorage.getItem('arDoencRenal') == null) || (localStorage.getItem('arDoencRenal') ==
'vazio')) {
        arDoencRenal.checked = false;
    } else {
        arDoencRenal.checked = true;
    }
}
retornaDadosCheckbox();

```

js/retonoDados/retornoDadosHemog.js

```

function retornaDadosHemograma(){
    let hemogHemac = document.querySelector('#hemog-hemac');
    let hemogHemogl = document.querySelector('#hemog-hemogl');
    let hemogHemat = document.querySelector('#hemog-hemat');
    let hemogHeuc = document.querySelector('#hemog-leuc');
    let hemogHinf = document.querySelector('#hemog-linf');
    let hemogHlaq = document.querySelector('#hemog-plaq');
    let hemogHlic = document.querySelector('#hemog-glic');
    let hemogColes = document.querySelector('#hemog-coles');

```

```

let hemogHdl = document.querySelector('#hemog-hdl');
let hemogTrig = document.querySelector('#hemog-trig');
let hemogHemogGlic = document.querySelector('#hemog-hemog-glic');
let hemogCreat = document.querySelector('#hemog-creat');
let hemogPotass = document.querySelector('#hemog-potass');
let hemogEqu = document.querySelector('#hemog-equ');
let hemogMicroa = document.querySelector('#hemog-microa');
let hemogAlbum = document.querySelector('#hemog-album');
let hemogAcid = document.querySelector('#hemog-acid');
let hemogUreia = document.querySelector('#hemog-ureia');
let hemogTgo = document.querySelector('#hemog-tgo');
let hemogTgp = document.querySelector('#hemog-tgp');

hemogHemac.value = localStorage.getItem('hemogHemac');
hemogHemogl.value = localStorage.getItem('hemogHemogl');
hemogHemat.value = localStorage.getItem('hemogHemat');
hemogHeuc.value = localStorage.getItem('hemogHeuc');
hemogHinf.value = localStorage.getItem('hemogHinf');
hemogHlaq.value = localStorage.getItem('hemogHlaq');
hemogHlic.value = localStorage.getItem('hemogHlic');
hemogColes.value = localStorage.getItem('hemogColes');
hemogHdl.value = localStorage.getItem('hemogHdl');
hemogTrig.value = localStorage.getItem('hemogTrig');
hemogHemogGlic.value = localStorage.getItem('hemogHemogGlic');
hemogCreat.value = localStorage.getItem('hemogCreat');
hemogPotass.value = localStorage.getItem('hemogPotass');
hemogEqu.value = localStorage.getItem('hemogEqu');
hemogMicroa.value = localStorage.getItem('hemogMicroa');
hemogAlbum.value = localStorage.getItem('hemogAlbum');
hemogAcid.value = localStorage.getItem('hemogAcid');
hemogUreia.value = localStorage.getItem('hemogUreia');
hemogTgo.value = localStorage.getItem('hemogTgo');
hemogTgp.value = localStorage.getItem('hemogTgp');

}
retornaDadosHemograma();

```

js/saveCadastros/cadEquipe.js

```

/*
const btnCadastrar = document.querySelector('#cadastrar');

function insertNovoUsuario(){
    let userCadNome = document.querySelector('#nome').value;
    let userCadData = document.querySelector('#dataNasc').value;
    let userCadCpf = document.querySelector('#cpf').value;
    let userCadCartao = document.querySelector('#cartSus').value;
    let userCadAgente = document.querySelector('#agenteSus').value;
    let userCadMasc = document.querySelector('#masc');
    let userCadFem = document.querySelector('#femi');

    if(userCadMasc.checked == true){
        localStorage.setItem('userCadSexo', userCadMasc.value);
    }else if(userCadFem.checked == true){
        localStorage.setItem('userCadSexo', userCadFem.value);
    }
}

```

```

    }

    localStorage.setItem('userCadNome', userCadNome);
    localStorage.setItem('userCadData', userCadData);
    localStorage.setItem('userCadCpf', userCadCpf);
    localStorage.setItem('userCadCartao', userCadCartao);
    localStorage.setItem('userCadAgente', userCadAgente);

    console.log('Sucesso Usuario:', userCadNome, userCadData,
    userCadCpf, userCadCartao, userCadAgente, localStorage.getItem('userCadSexo'));

}

btnCadastrar.addEventListener("click", insertNovoUsuario);
*/

js/saveCadastros/cadUsuario.js

const btnCadastrar = document.querySelector('#cadastrar');

function insertNovoUsuario(){
    let userCadNome = document.querySelector('#nome').value;
    let userCadData = document.querySelector('#dataNasc').value;
    let userCadCpf = document.querySelector('#cpf').value;
    let userCadCartao = document.querySelector('#cartSus').value;
    let userCadAgente = document.querySelector('#agenteSus').value;
    let userCadMasc = document.querySelector('#masc');
    let userCadFem = document.querySelector('#femi');

    if(userCadMasc.checked == true){
        localStorage.setItem('userCadSexo', userCadMasc.value);

    }else if(userCadFem.checked == true){
        localStorage.setItem('userCadSexo', userCadFem.value);

    }

    localStorage.setItem('userCadNome', userCadNome);
    localStorage.setItem('userCadData', userCadData);
    localStorage.setItem('userCadCpf', userCadCpf);
    localStorage.setItem('userCadCartao', userCadCartao);
    localStorage.setItem('userCadAgente', userCadAgente);

    console.log('Sucesso Usuario:', userCadNome, userCadData,
    userCadCpf, userCadCartao, userCadAgente, localStorage.getItem('userCadSexo'));

}

btnCadastrar.addEventListener("click", insertNovoUsuario);

js/saveStorageNhb/amoraceitacao.js

const btnSalvar = document.querySelector('#btn-amoraceitacao-salvar');

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-amo', checks.length);

```



```

    for(let i=0; i<checks.length; i++){
      if(checks[i].checked == true){
        let name = `amo-${i}`;
        localStorage.setItem(name, checks[i].name);
      }
    }
  }
}

function insertChecks(){
  let el = document.querySelector('#retorno-amoraceitacao');
  console.log("Salvando...");
  save();
  el.innerHTML = "Salvo com sucesso";
  setTimeout(function(){
    //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
  }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/atividadefisica.js

```

```

const btnSalvar = document.querySelector('#btn-atividadefisica-salvar');

```

```

function save(){
  let checks = document.querySelectorAll("div > input");
  localStorage.setItem('qtd-fis', checks.length);
  for(let i=0; i<checks.length; i++){
    if(checks[i].checked == true){
      let name = `fis-${i}`;
      localStorage.setItem(name, checks[i].name);
    }
  }
}

```

```

function insertChecks(){
  let el = document.querySelector('#retorno-atividadefisica');
  console.log("Salvando...");
  save();
  el.innerHTML = "Salvo com sucesso";
  setTimeout(function(){
    //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
  }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/educacao.js

```

```

const btnSalvar = document.querySelector('#btn-educacao-salvar');

```

```

function save(){
  let checks = document.querySelectorAll("div > input");
  localStorage.setItem('qtd-edu', checks.length);
  for(let i=0; i<checks.length; i++){
    if(checks[i].checked == true){

```

```

        let name = `edu-${i}`;
        localStorage.setItem(name, checks[i].name);
    }
}
}

function insertChecks(){
    let el = document.querySelector('#retorno-educacao');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

js/saveStorageNhb/liberdade.js

```

const btnSalvar = document.querySelector('#btn-liberdade-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-lib', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `lib-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-liberdade');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

js/saveStorageNhb/nutricao.js

```

const btnSalvar = document.querySelector('#btn-nutricao-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-nut', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `nut-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

    }
}

function insertChecks(){
    let el = document.querySelector('#retorno-nutricao');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/oxigenacao.js

```

```

const btnSalvar = document.querySelector('#btn-oxigenacao-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-oxi', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `oxi-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-oxigenacao');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/regulacao hormonal.js

```

```

const btnSalvar = document.querySelector('#btn-regulacao hormonal-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-hor', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `hor-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
  let el = document.querySelector('#retorno-regulacaohormonal');
  console.log("Salvando...");
  save();
  el.innerHTML = "Salvo com sucesso";
  setTimeout(function(){
    //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
  }, 2000);
}

btnSalvar.addEventListener("click", insertChecks);

js/saveStorageNhb/regulacaoneurologica.js

const btnSalvar = document.querySelector('#btn-regulacaoneurologica-salvar');

function save(){
  let checks = document.querySelectorAll("div > input");
  localStorage.setItem('qtd-neu', checks.length);
  for(let i=0; i<checks.length; i++){
    if(checks[i].checked == true){
      let name = `neu-${i}`;
      localStorage.setItem(name, checks[i].name);
    }
  }
}

function insertChecks(){
  let el = document.querySelector('#retorno-regulacaoneurologica');
  console.log("Salvando...");
  save();
  el.innerHTML = "Salvo com sucesso";
  setTimeout(function(){
    //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
  }, 2000);
}

btnSalvar.addEventListener("click", insertChecks);

js/saveStorageNhb/regulacaovascular.js

const btnSalvar = document.querySelector('#btn-regulacaovascular-salvar');

function save(){
  let checks = document.querySelectorAll("div > input");
  localStorage.setItem('qtd-vas', checks.length);
  for(let i=0; i<checks.length; i++){
    if(checks[i].checked == true){
      let name = `vas-${i}`;
      localStorage.setItem(name, checks[i].name);
    }
  }
}

function insertChecks(){
  let el = document.querySelector('#retorno-regulacaovascular');
  console.log("Salvando...");

```

```

    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/religiosidade.js

```

```

const btnSalvar = document.querySelector('#btn-religiosidade-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-reg', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `reg-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-religiosidade');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/sentidos.js

```

```

const btnSalvar = document.querySelector('#btn-sentidos-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-sen', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `sen-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-sentidos');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){

```

```

        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/sonorepouso.js

```

```

const btnSalvar = document.querySelector('#btn-sonoerepouso-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-son', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `son-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-sonorepouso');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```

btnSalvar.addEventListener("click", insertChecks);

```

```

js/saveStorageNhb/terapeutica.js

```

```

const btnSalvar = document.querySelector('#btn-terapeutica-salvar');

```

```

function save(){
    let checks = document.querySelectorAll("div > input");
    localStorage.setItem('qtd-ter', checks.length);
    for(let i=0; i<checks.length; i++){
        if(checks[i].checked == true){
            let name = `ter-${i}`;
            localStorage.setItem(name, checks[i].name);
        }
    }
}

```

```

function insertChecks(){
    let el = document.querySelector('#retorno-terapeutica');
    console.log("Salvando...");
    save();
    el.innerHTML = "Salvo com sucesso";
    setTimeout(function(){
        //window.location = 'http://localhost:5500/pages/pageDiagnosEnf.html';
    }, 2000);
}

```

```
    }, 2000);  
}
```

```
btnSalvar.addEventListener("click", insertChecks);
```

js/Script imprimir.js

```
function downloadPDF() {  
    const item = document.querySelector(".conteudo_relatorio");  
  
    var opt = {  
        margin: 0,  
        filename: "myfile.pdf",  
        html2canvas: { scale: 2 },  
        jsPDF: { unit: "in", format: "letter", orientation: "portrait" },  
    };  
  
    html2pdf().set(opt).from(item).save();  
}
```

js/ajudaUser.js

```
const btnAjuda = document.querySelector('#btn-ajuda');  
  
function ajudaUser(){  
    const el = document.querySelector('#retorno-ajuda-user');  
    el.innerHTML = 'https://www.youtube.com/channel/UCSHtFhjmheShsGOyvWqdVjA';  
    el.style.fontSize = '1rem';  
    el.style.color = 'green';  
    el.style.fontWeight = '500';  
    setTimeout(function(){  
        el.innerHTML = "";  
  
        }, 7000);  
}  
btnAjuda.addEventListener("click", ajudaUser);
```

js/configNivelUser.js

```
const btnInic = document.querySelector('#btn-nivel-user-inic');  
const btnExper = document.querySelector('#btn-nivel-user-exper');  
const btnLeft = document.querySelector('.disable-btn-left');  
  
const footer = document.querySelector('.disable-footer');  
const btnBusca = document.querySelector('.disable-btn-busca');  
const btnConsulta = document.querySelector('.disable-btn-consulta');  
const btnProtocol = document.querySelector('.disable-btn-protocol');  
const nivelUserIni = document.querySelector('.nivel-user-ini');  
const nivelUserExper = document.querySelector('.nivel-user-exper');  
const btnRelatorio = document.querySelector('.disable-btn-relatorio');  
  
function iniciante(){  
    const el = document.querySelector('#retorno-config-nivel-user');  
    el.innerHTML = 'Configuração salva, Aguarde!!';  
    el.style.fontSize = '1rem';
```

```

el.style.color = 'black';
el.style.fontWeight = '500';
setTimeout(function(){
    el.innerHTML = '';
    btnBusca.style.display = 'block';
    btnConsulta.style.display = 'block';
    btnRelatorio.style.display = 'block';
    nivelUserIni.style.display = 'none';
    nivelUserExper.style.display = 'none';
    btnLeft.style.cssText = 'display: none !important';

}, 3000);
localStorage.setItem('ConfigNivelUser', 'iniciante');

}
function experiente(){
    const el = document.querySelector('#retorno-config-nivel-user');
    el.innerHTML = 'Configuração salva, Aguarde!!';
    el.style.fontSize = '1rem';
    el.style.color = 'black';
    el.style.fontWeight = '500';
    setTimeout(function(){
        el.innerHTML = '';
        btnBusca.style.display = 'block';
        btnConsulta.style.display = 'block';
        btnProtocol.style.display = 'block';
        btnRelatorio.style.display = 'block';
        nivelUserIni.style.display = 'none';
        nivelUserExper.style.display = 'none';
        btnLeft.style.cssText = 'display: flex !important';

    }, 3000);
    localStorage.setItem('ConfigNivelUser', 'experiente');

}

btnInic.addEventListener("click", iniciante);
btnExper.addEventListener("click", experiente);

```

js/enableAdmin.js

```

function enabledAdmin(){
    let enabledCaseAdmin = document.querySelector('#enabled-case-admin');
    //enabledCaseAdmin.style.display = 'block';
    if(localStorage.getItem('ehAdmin') == 'ehAdmin'){
        enabledCaseAdmin.style.display = 'block';
        console.log('É admin: ', localStorage.getItem('ehAdmin'));

    }else if((localStorage.getItem('ehAdmin') == 'naoEhAdmin') || (localStorage.getItem('ehAdmin') == null)){
        enabledCaseAdmin.style.display = 'none';
        console.log('Não é admin: ', localStorage.getItem('ehAdmin'));

    }
}

```



```
enabledAdmin();
```

```
js/finalizaAvaClinic.js
```

```
const btnAvaClinic = document.querySelector('#btn-finaliza-ava-clinic');  
const retornoFinalizaAvaClinic = document.querySelector('#retorno-finaliza-ava-clinic');
```

```
function finalizaAvaClinic() {  
  retornoFinalizaAvaClinic.innerHTML = 'Salvando dados. Aguarde...';  
  retornoFinalizaAvaClinic.style.color = 'green';  
  retornoFinalizaAvaClinic.style.fontSize = '1.2rem';
```

```
  setTimeout(function () {  
    retornoFinalizaAvaClinic.innerHTML = 'Dados salvos com sucesso!!';  
    retornoFinalizaAvaClinic.style.color = 'green';
```

```
  }, 2000);
```

```
  setTimeout(function () {  
    retornoFinalizaAvaClinic.innerHTML = '...';  
    retornoFinalizaAvaClinic.style.color = 'black';  
    console.log('Dados avaliação clinica salvos');  
    window.location = 'pageDiagnosEnf.html'
```

```
  }, 4000);
```

```
}
```

```
btnAvaClinic.addEventListener("click", finalizaAvaClinic);
```

```
js/finalizaConsulta.js
```

```
const btnFinalizaConsulta = document.querySelector('#btn-finaliza-consulta');  
const retornoFinalizaCosulta = document.querySelector('#retorno-finaliza-cosulta');  
const saveUser = localStorage.getItem('ehAdmin');  
const idEquipe = localStorage.getItem('id_equipe');  
const idUserSus = localStorage.getItem('idUserSus');  
const nivelUser = localStorage.getItem('ConfigNivelUser');
```

```
function finalizaConsulta() {  
  retornoFinalizaCosulta.innerHTML = 'Salvando dados. Aguarde... Para visualizar e imprimir o  
atendimento realizado retorne ao MENU INICIAL, selecione a opção (BUSCAR USUÁRIO(A)/RELATÓRIO  
INDIVIDUAL, digite o cartão SUS);  
  retornoFinalizaCosulta.style.color = 'green';  
  retornoFinalizaCosulta.style.fontSize = '1.2rem';
```

```
  setTimeout(function () {  
    retornoFinalizaCosulta.innerHTML = 'Dados salvos com sucesso!! Para visualizar e imprimir o  
atendimento realizado retorne ao MENU INICIAL, selecione a opção (BUSCAR USUÁRIO(A)/RELATÓRIO  
INDIVIDUAL, digite o cartão SUS);  
    retornoFinalizaCosulta.style.color = 'green';  
    localStorage.clear();  
    console.log('# LocalStorage Limpo #');
```

```
  }, 10000);
```

```
  setTimeout(function () {  
    retornoFinalizaCosulta.innerHTML = '...';
```

```

    retornoFinalizaConsulta.style.color = 'black';
    localStorage.setItem('ehAdmin', saveUser);
    localStorage.setItem('id_equipe', idEquipe);
    localStorage.setItem('idUserSus', idUserSus);
    localStorage.setItem('ConfigNivelUser', nivelUser);
    console.log('--> ', localStorage.getItem('ehAdmin'));
    console.log('Id Equipe: ', localStorage.getItem('id_equipe'));
    console.log('Id User Sus: ', localStorage.getItem('idUserSus'));
    console.log('Nivel user: ', localStorage.getItem('ConfigNivelUser'));

    window.location = "../pages/adm_area.html";

    }, 5000);

}

btnFinalizaConsulta.addEventListener("click", finalizaConsulta);

js/insertDadosStorageAvaClinic.js

const btnPasPad = document.querySelector('#btn-calcular-pas-pad');
const btnImc = document.querySelector('#btn-imc-calcular');
const btnCircAbdom = document.querySelector('#btn-circ-abdom-calcular');
const btnCintQuadr = document.querySelector('#btn-cint-quadr-calcular');
const btnTornBraq = document.querySelector('#btn-torn-braq-calcular');

const btnEstratCalcular = document.querySelector('#btn-estrat-calcular');
const btnEstrat2Calcular = document.querySelector('#btn-estrat2-calcular');

const etapa2PaSistol = document.querySelector('#etapa2-pa-sistol');
const tornBraqPasBra = document.querySelector('#torn-braq-pas-bra');

function insertPasPad() {
    let pad = document.querySelector('#pad').value;
    let pas = document.querySelector('#pas').value;

    localStorage.setItem('pad', pad);
    localStorage.setItem('pas', pas);

    etapa2PaSistol.value = localStorage.getItem('pas');
    tornBraqPasBra.value = localStorage.getItem('pas');
    console.log('Sucesso Pas-Pad:', pas, pad);
}

function insertImc() {
    let peso = document.querySelector('#imc-peso').value;
    let altura = document.querySelector('#imc-altura').value;
    altura = altura.replace(".", ".");

    localStorage.setItem('peso', peso);
    localStorage.setItem('altura', altura);

    console.log('Sucesso Peso-Altura:', peso, altura);
    calculaImc(peso, altura);
    console.log(calculaImc);
}

```

```

}
function insertCircAbdom() {
    let circAbdomCm = document.querySelector('#circ-abdom-cm').value;

    localStorage.setItem('circAbdomCm', circAbdomCm);

    console.log('Sucesso Circunferencia Abdominal:', circAbdomCm);
}
function insertCintQuadr() {
    let cintQuadrAbdomCm = document.querySelector('#cint-quadr-abdom-cm').value;
    let cintQuadrQuadrCm = document.querySelector('#cint-quadr-quadr-cm').value;

    localStorage.setItem('cintQuadrAbdomCm', cintQuadrAbdomCm);
    localStorage.setItem('cintQuadrQuadrCm', cintQuadrQuadrCm);

    console.log('Sucesso Cintura-Quadril:', cintQuadrAbdomCm, cintQuadrQuadrCm);
}
function insertTornBraq() {
    let tornBraqPasTorn = document.querySelector('#torn-braq-pas-torn').value;
    let tornBraqPasBra = document.querySelector('#torn-braq-pas-bra').value;

    localStorage.setItem('tornBraqPasTorn', tornBraqPasTorn);
    localStorage.setItem('pas', tornBraqPasBra);

    etapa2PaSistol.value = localStorage.getItem('pas');
    pas.value = localStorage.getItem('pas');

    console.log('Sucesso Tornozelo-Braquial:', tornBraqPasTorn, tornBraqPasBra);
}
function insertEstratificacao() {
    /*Variaveis Risco Baixo*/
    let brIdade = document.querySelector('#br-idade');
    let brSexMasc = document.querySelector('#br-sex-masc');
    let brTabag = document.querySelector('#br-tabag');
    let brHipert = document.querySelector('#br-hipert');
    let brObs = document.querySelector('#br-obs');
    let brSedent = document.querySelector('#br-sedent');
    let brHistoFam = document.querySelector('#br-histo-fam');
    let brHistoDoenca = document.querySelector('#br-histo-doenca');
    let brManifest = document.querySelector('#br-manifest');
    let brDiagnDislip = document.querySelector('#br-diagn-dislip');
    let brDiagnPolic = document.querySelector('#br-diagn-polic');

    /*Variaveis Risco Alto */
    let arAcidVasc = document.querySelector('#ar-acid-vasc');
    let arInfartAgud = document.querySelector('#ar-infart-agud');
    let arLesPerif = document.querySelector('#ar-les-perif');
    let arAtaqIsqu = document.querySelector('#ar-ataq-isqu');
    let arHiperVentr = document.querySelector('#ar-hiper-ventr');
    let arNefrop = document.querySelector('#ar-nefrop');
    let arRetin = document.querySelector('#ar-retin');
    let arAneur = document.querySelector('#ar-aneur');
    let arEsten = document.querySelector('#ar-esten');
    let arDiabet = document.querySelector('#ar-diabet');

```

```

let arDoenVasc = document.querySelector('#ar-doen-vasc');
let arInsuf = document.querySelector('#ar-insuf');
let arAngina = document.querySelector('#ar-angina');
let arDoencRenal = document.querySelector('#ar-doenc-renal');

/*Variavel da etapa 2*/
let checkFumante = document.querySelector('#etapa2-fumante');
checkFumante.checked = false;

let contBaixoRisco = 0;
let contAltoRisco = 0;

/*Condição Risco Baixo*/
if (brIdade.checked == true) {
    brIdade = document.querySelector('#br-idade').name;
    contBaixoRisco++;
} else {
    brIdade = "vazio"
}

if (brSexMasc.checked == true) {
    brSexMasc = document.querySelector('#br-sex-masc').name;
    contBaixoRisco++;
} else {
    brSexMasc = "vazio"
}

if (brTabag.checked == true) {
    brTabag = document.querySelector('#br-tabag').name;
    checkFumante.checked = true;
    contBaixoRisco++;
} else {
    brTabag = "vazio"
}

if (brHipert.checked == true) {
    brHipert = document.querySelector('#br-hipert').name;
    contBaixoRisco++;
} else {
    brHipert = "vazio"
}

if (brObs.checked == true) {
    brObs = document.querySelector('#br-obs').name;
    contBaixoRisco++;
} else {
    brObs = "vazio"
}

if (brSedent.checked == true) {
    brSedent = document.querySelector('#br-sedent').name;
    contBaixoRisco++;
} else {
    brSedent = "vazio"
}

if (brHistoFam.checked == true) {

```

```

    brHistoFam = document.querySelector('#br-histo-fam').name;
    contBaixoRisco++;
} else {
    brHistoFam = "vazio"
}

if (brHistoDoenca.checked == true) {
    brHistoDoenca = document.querySelector('#br-histo-doenca').name;
    contBaixoRisco++;
} else {
    brHistoDoenca = "vazio"
}

if (brManifest.checked == true) {
    brManifest = document.querySelector('#br-manifest').name;
    contBaixoRisco++;
} else {
    brManifest = "vazio"
}

if (brDiagnDislip.checked == true) {
    brDiagnDislip = document.querySelector('#br-diagn-dislip').name;
    contBaixoRisco++;
} else {
    brDiagnDislip = "vazio"
}

if (brDiagnPolic.checked == true) {
    brDiagnPolic = document.querySelector('#br-diagn-polic').name;
    contBaixoRisco++;
} else {
    brDiagnPolic = "vazio"
}

/*Condição Risco Alto*/
if (arAcidVasc.checked == true) {
    arAcidVasc = document.querySelector('#ar-acid-vasc').name;
    contAltoRisco++;
} else {
    arAcidVasc = "vazio"
}

if (arInfartAgud.checked == true) {
    arInfartAgud = document.querySelector('#ar-infart-agud').name;
    contAltoRisco++;
} else {
    arInfartAgud = "vazio"
}

if (arLesPerif.checked == true) {
    arLesPerif = document.querySelector('#ar-les-perif').name;
    contAltoRisco++;
} else {
    arLesPerif = "vazio"
}

if (arAtaqlsqu.checked == true) {

```

```
    arAtaqlsqu = document.querySelector('#ar-ataq-isqu').name;
    contAltoRisco++;
} else {
    arAtaqlsqu = "vazio"
}

if (arHiperVentr.checked == true) {
    arHiperVentr = document.querySelector('#ar-hiper-ventr').name;
    contAltoRisco++;
} else {
    arHiperVentr = "vazio"
}

if (arNefrop.checked == true) {
    arNefrop = document.querySelector('#ar-nefrop').name;
    contAltoRisco++;
} else {
    arNefrop = "vazio"
}

if (arRetin.checked == true) {
    arRetin = document.querySelector('#ar-retin').name;
    contAltoRisco++;
} else {
    arRetin = "vazio"
}

if (arAneur.checked == true) {
    arAneur = document.querySelector('#ar-aneur').name;
    contAltoRisco++;
} else {
    arAneur = "vazio"
}

if (arEsten.checked == true) {
    arEsten = document.querySelector('#ar-esten').name;
    contAltoRisco++;
} else {
    arEsten = "vazio"
}

if (arDiabet.checked == true) {
    arDiabet = document.querySelector('#ar-diabet').name;
    contAltoRisco++;
} else {
    arDiabet = "vazio"
}

if (arDoenVasc.checked == true) {
    arDoenVasc = document.querySelector('#ar-doen-vasc').name;
    contAltoRisco++;
} else {
    arDoenVasc = "vazio"
}

if (arInsuf.checked == true) {
    arInsuf = document.querySelector('#ar-insuf').name;
```

```

        contAltoRisco++;
    } else {
        arInsuf = "vazio"
    }

    if (arAngina.checked == true) {
        arAngina = document.querySelector('#ar-angina').name;
        contAltoRisco++;
    } else {
        arAngina = "vazio"
    }

    if (arDoencRenal.checked == true) {
        arDoencRenal = document.querySelector('#ar-doenc-renal').name;
        contAltoRisco++;
    } else {
        arDoencRenal = "vazio"
    }

    console.log('Sucesso Estratificação Baixo:', brIdade, brSexMasc, brTabag, brHipert, brObs,
        brSedent, brHistoFam, brHistoDoenca, brManifest,
        brDiagnDislip, brDiagnPolic);

    console.log('Sucesso Estratificação Alto:', arAcidVasc, arInfartAgud, arLesPerif,
        arAtaqlsqu, arHiperVentr, arNefrop, arRetin,
        arAneur, arEsten, arDiabet, arDoenVasc, arInsuf,
        arAngina, arDoencRenal);

    verificaRisco(contBaixoRisco, contAltoRisco);

    /*Salva Baixo Risco em LocalStorage */
    localStorage.setItem('brIdade', brIdade);
    localStorage.setItem('brSexMasc', brSexMasc);
    localStorage.setItem('brTabag', brTabag);
    localStorage.setItem('brHipert', brHipert);
    localStorage.setItem('brObs', brObs);
    localStorage.setItem('brSedent', brSedent);
    localStorage.setItem('brHistoFam', brHistoFam);
    localStorage.setItem('brHistoDoenca', brHistoDoenca);
    localStorage.setItem('brManifest', brManifest);
    localStorage.setItem('brDiagnDislip', brDiagnDislip);
    localStorage.setItem('brDiagnPolic', brDiagnPolic);

    /*Salva Baixo Risco em LocalStorage */
    localStorage.setItem('arAcidVasc', arAcidVasc);
    localStorage.setItem('arInfartAgud', arInfartAgud);
    localStorage.setItem('arLesPerif', arLesPerif);
    localStorage.setItem('arAtaqlsqu', arAtaqlsqu);
    localStorage.setItem('arHiperVentr', arHiperVentr);
    localStorage.setItem('arNefrop', arNefrop);
    localStorage.setItem('arRetin', arRetin);
    localStorage.setItem('arAneur', arAneur);
    localStorage.setItem('arEsten', arEsten);
    localStorage.setItem('arDiabet', arDiabet);
    localStorage.setItem('arDoenVasc', arDoenVasc);
    localStorage.setItem('arInsuf', arInsuf);

```

```

localStorage.setItem('arAngina', arAngina);
localStorage.setItem('arDoencRenal', arDoencRenal);

}

function insertEstratificacaoEtapa2() {
  let etapa2ColestTotal = document.querySelector('#etapa2-colest-total').value;
  let etapa2Fumante = document.querySelector('#etapa2-fumante');
  let etapa2Hdl = document.querySelector('#etapa2-hdl').value;
  let etapa2PaSistol = document.querySelector('#etapa2-pa-sistol').value;
  let etapa2Tratada = document.querySelector('#etapa2-tratada');

  if (etapa2Fumante.checked == true) {
    localStorage.setItem('brTabag', etapa2Fumante.name);
    //localStorage.setItem('etapa2Fumante', etapa2Fumante.name);

  } else {
    localStorage.setItem('brTabag', 'Não fumante');

  }

  if (etapa2Tratada.checked == true) {
    localStorage.setItem('etapa2Tratada', etapa2Tratada.name);

  } else {
    localStorage.setItem('etapa2Tratada', 'Não tratada');

  }

  localStorage.setItem('hemogColes', etapa2ColestTotal);
  localStorage.setItem('hemogHdl', etapa2Hdl);
  localStorage.setItem('pas', etapa2PaSistol);

  tornBraqPasBra.value = localStorage.getItem('pas');
  pas.value = localStorage.getItem('pas');

  console.log(etapa2ColestTotal, localStorage.getItem('brTabag'), etapa2Hdl,
    etapa2PaSistol, localStorage.getItem('etapa2Tratada'));

}

function calculaImc(peso, altura) {
  let el = document.querySelector('#retorno-calculo-imc');
  let resultadoimc = document.querySelector('#retorno-calculo-imc2')
  let result = parseInt(peso) / (parseFloat(altura) * parseFloat(altura));
  console.log(result);
  resultadoimc.innerHTML = result.toFixed(2);
  let checkimc = document.querySelector('#br-obs');
  if (result > 30) {
    checkimc.checked = true;
  }
  else {
    checkimc.checked = false;
  }

  localStorage.setItem('imc', result);

  if (result < 18.5) {

```



```

        localStorage.setItem('clas_imc', 'Abaixo do peso');
        el.innerHTML = 'Abaixo do peso'

    } else if ((result >= 18.5) && (result < 25)) {
        localStorage.setItem('clas_imc', 'Peso normal');
        el.innerHTML = 'Peso normal'

    } else if ((result >= 25) && (result < 30)) {
        localStorage.setItem('clas_imc', 'Excesso de peso');
        el.innerHTML = 'Excesso de peso'

    } else if ((result >= 30) && (result < 35)) {
        localStorage.setItem('clas_imc', 'Obesidade classe I');
        el.innerHTML = 'Obesidade classe I'

    } else if ((result >= 35) && (result < 40)) {
        localStorage.setItem('clas_imc', 'Obesidade classe II');
        el.innerHTML = 'Obesidade classe II'

    } else if (result >= 40) {
        localStorage.setItem('clas_imc', 'Obesidade classe III');
        el.innerHTML = 'Obesidade classe III'

    }
}

function verificaRisco(contBaixoRisco, contAltoRisco) {
    let el = document.querySelector('#retorno-calculo-estrat');
    let el2 = document.querySelector('.disable-etapa-2');
    let el3 = document.querySelector('.disable-etapa-3');
    el.innerHTML = 'INDEFINIDO'
    el2.style.display = 'none';
    el3.style.display = 'none';

    if (contAltoRisco >= 1) {
        localStorage.setItem('clas_estrat', 'ALTO RISCO');
        el.innerHTML = 'ALTO RISCO';
        el2.style.display = 'none';
        el3.style.display = 'none';

    } else {
        if (contBaixoRisco == 1) {
            localStorage.setItem('clas_estrat', 'BAIXO RISCO');
            el.innerHTML = 'BAIXO RISCO';

        }

        if (contBaixoRisco >= 2) {
            el.innerHTML = 'Risco Indefinido! Prossiga para a Etapa 2 ...'
            el2.style.display = 'flex';
            el3.style.display = 'block';

        }

    }
}

btnPasPad.addEventListener("click", insertPasPad);
btnImc.addEventListener("click", insertImc);
btnEstratCalcular.addEventListener("click", insertEstratificacao);

```

```
btnCircAbdom.addEventListener("click", insertCircAbdom);
btnCintQuadr.addEventListener("click", insertCintQuadr);
btnTornBraq.addEventListener("click", insertTornBraq);
btnEstrat2Calcular.addEventListener('click', insertEstratificacaoEtapa2);
```

js/insertDadosStorageHemog.js

```
const btnHemogCalcular = document.querySelector('#btn-hemog-calcular');
```

```
function insertHemograma() {
  let hemogHemac = document.querySelector('#hemog-hemac').value;
  let hemogHemogl = document.querySelector('#hemog-hemogl').value;
  let hemogHemat = document.querySelector('#hemog-hemat').value;
  let hemogHeuc = document.querySelector('#hemog-leuc').value;
  let hemogHinf = document.querySelector('#hemog-linf').value;
  let hemogHlaq = document.querySelector('#hemog-plaq').value;
  let hemogHlic = document.querySelector('#hemog-glic').value;
  let hemogColes = document.querySelector('#hemog-coles').value;
  let hemogHdl = document.querySelector('#hemog-hdl').value;
  let hemogTrig = document.querySelector('#hemog-trig').value;
  let hemogHemogGlic = document.querySelector('#hemog-hemog-glic').value;
  let hemogCreat = document.querySelector('#hemog-creat').value;
  let hemogPotass = document.querySelector('#hemog-potass').value;
  let hemogEqu = document.querySelector('#hemog-equ').value;
  let hemogMicroa = document.querySelector('#hemog-microa').value;
  let hemogAlbum = document.querySelector('#hemog-album').value;
  let hemogAcid = document.querySelector('#hemog-acid').value;
  let hemogUreia = document.querySelector('#hemog-ureia').value;
  let hemogTgo = document.querySelector('#hemog-tgo').value;
  let hemogTgp = document.querySelector('#hemog-tgp').value;
```

```
  localStorage.setItem('hemogHemac', hemogHemac);
  localStorage.setItem('hemogHemogl', hemogHemogl);
  localStorage.setItem('hemogHemat', hemogHemat);
  localStorage.setItem('hemogHeuc', hemogHeuc);
  localStorage.setItem('hemogHinf', hemogHinf);
  localStorage.setItem('hemogHlaq', hemogHlaq);
  localStorage.setItem('hemogHlic', hemogHlic);
  localStorage.setItem('hemogColes', hemogColes);
  localStorage.setItem('hemogHdl', hemogHdl);
  localStorage.setItem('hemogTrig', hemogTrig);
  localStorage.setItem('hemogHemogGlic', hemogHemogGlic);
  localStorage.setItem('hemogCreat', hemogCreat);
  localStorage.setItem('hemogPotass', hemogPotass);
  localStorage.setItem('hemogEqu', hemogEqu);
  localStorage.setItem('hemogMicroa', hemogMicroa);
  localStorage.setItem('hemogAlbum', hemogAlbum);
  localStorage.setItem('hemogAcid', hemogAcid);
  localStorage.setItem('hemogUreia', hemogUreia);
  localStorage.setItem('hemogTgo', hemogTgo);
  localStorage.setItem('hemogTgp', hemogTgp);
```

```
  console.log('Sucesso Hemograma:', hemogHemac, hemogHemogl, hemogHemat, hemogHeuc,
    hemogHinf, hemogHlaq, hemogHlic, hemogColes, hemogHdl, hemogTrig,
    hemogHemogGlic, hemogCreat, hemogPotass, hemogEqu, hemogMicroa, hemogAlbum,
    hemogAcid, hemogUreia, hemogTgo, hemogTgp);
```

```

let hemogramaSalvo = document.querySelector('#retorno-salvo-hemograma');
hemogramaSalvo.innerHTML = 'Salvo com sucesso';
hemogramaSalvo.style.color = 'green';
setTimeout(function () {
    hemogramaSalvo.innerHTML = '...';
    window.location = 'pageAvaClinic.html'

}, 3000);

}

```

```

btnHemogCalcular.addEventListener("click", insertHemograma);

```

js/logoutSistema.js

```

const btnLogout = document.querySelector('.btn-logout');

```

```

function logoutClean(){
    localStorage.clear();
    console.log('Logout do sistema!!');
}

```

```

btnLogout.addEventListener('click', logoutClean);

```

js/verificaNivelUser.js

```

function disableConfigNiveis(){
    const btnBusca = document.querySelector('.disable-btn-busca');
    const btnConsulta = document.querySelector('.disable-btn-consulta');
    const btnProtocol = document.querySelector('.disable-btn-protocol');
    const nivelUserIni = document.querySelector('.nivel-user-ini');
    const nivelUserExper = document.querySelector('.nivel-user-exper');
    const btnLeft = document.querySelector('.disable-btn-left');
    const btnRelatorio = document.querySelector('.disable-btn-relatorio');

```

```

    if(localStorage.getItem('ehAdmin') == 'ehAdmin'){
        btnBusca.style.display = 'none';
        btnConsulta.style.display = 'none';
        btnProtocol.style.display = 'block';
        btnRelatorio.style.display = 'none';
        nivelUserIni.style.display = 'none';
        nivelUserExper.style.display = 'none';
        btnLeft.style.cssText = 'display: flex !important';
        console.log('User experiente/ admin');
    }

```

```

    }else{
        if(localStorage.getItem('ConfigNivelUser') == 'experiente'){
            btnBusca.style.display = 'block';
            btnConsulta.style.display = 'block';
            btnProtocol.style.display = 'block';
            btnRelatorio.style.display = 'block';
            nivelUserIni.style.display = 'none';
            nivelUserExper.style.display = 'none';
            btnLeft.style.cssText = 'display: flex !important';
            console.log('User experiente');
        }
    }
}

```

```

    }else if(localStorage.getItem('ConfigNivelUser') == 'iniciante'){
        btnBusca.style.display = 'block';
        btnConsulta.style.display = 'block';
        btnRelatorio.style.display = 'block';
        nivelUserIni.style.display = 'none';
        nivelUserExper.style.display = 'none';
        btnLeft.style.cssText = 'display: none !important';
        console.log('User iniciante');

    }else{
        btnBusca.style.display = 'none';
        btnConsulta.style.display = 'none';
        btnProtocol.style.display = 'none';
        btnRelatorio.style.display = 'none';
        nivelUserIni.style.display = 'block';
        nivelUserExper.style.display = 'block';
        btnLeft.style.cssText = 'display: none !important';
        console.log('User não informado');

    }
}
}
}
disableConfigNiveis();

```

js/verificaNivelUserFooter.js

```

const footer = document.querySelector('.disable-footer');
const fundo1 = document.querySelector('.fundo1');
const btnLeft = document.querySelector('.disable-btn-left');

function disableFooter(){
    if(localStorage.getItem('ehAdmin') == 'ehAdmin'){
        footer.style.display = 'none';
        fundo1.style.justifyContent = 'flex-start';
        btnLeft.style.cssText = 'display: flex !important';

    }else{
        if(localStorage.getItem('ConfigNivelUser') == 'experiente'){
            footer.style.display = 'none';
            fundo1.style.justifyContent = 'flex-start';
            btnLeft.style.cssText = 'display: flex !important';

        }else if(localStorage.getItem('ConfigNivelUser') == 'iniciante'){
            footer.style.display = 'flex';
            btnLeft.style.cssText = 'display: none !important';

        }else{
            footer.style.display = 'none';
            fundo1.style.justifyContent = 'flex-start';
            btnLeft.style.cssText = 'display: none !important';

        }
    }
}
disableFooter();

```

pages/pagesNecessidades/pageNecessBiolog.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="../../cadastro.html">
                    
                    <p class="text-menu">SAIR</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>
</body>
</html>
```

```

</nav>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="/pageNecessBiolog.html">
                
                <p class="text-menu"> NECESSIDADES PSICOBIOLOGICAS </p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/pageNecessSocia.html">
                
                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/pageNecessEspir.html">
                
                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
  <div>
    <div class="fundo-botao">
      <a href="/necessidades/oxigenacao.html" class="botao-padrao">
        <button>NECESSIDADES DE OXIGENAÇÃO</button>
      </a>
      <a href="/necessidades/nutricao.html" class="botao-padrao">
        <button>NECESSIDADES DE NUTRIÇÃO</button>
      </a>
      <a href="/necessidades/repousoesono.html" class="botao-padrao">
        <button>NECESSIDADES DE SONO E REPOUSO</button>
      </a>
      <a href="/necessidades/atividadefisica.html" class="botao-padrao">
        <button>NECESSIDADES DE ATIVIDADE FÍSICA</button>
      </a>
      <a href="/necessidades/regulacaovascular.html" class="botao-padrao">
        <button>NECESSIDADES DE REGULAÇÃO VASCULAR</button>
      </a>
      <a href="/necessidades/regulacaohormonal.html" class="botao-padrao">
        <button>NECESSIDADES DE REGULAÇÃO HORMONAL</button>
      </a>
    </div>
  </div>

```

```

<a href="/necessidades/neurologica.html" class="botao-padrao">
  <button>NECESSIDADES DE REGULAÇÃO NEUROLÓGICA</button>
</a>
<a href="/necessidades/sentido.html" class="botao-padrao">
  <button>NECESSIDADES DE ÓRGÃOS DO SENTIDO</button>
</a>
<a href="/necessidades/terapeutica.html" class="botao-padrao">
  <button>NECESSIDADES DE TERAPÊUTICA</button>
</a>
</div>
</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/pageNecessEspir.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>

```

```

<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="../menu2.html">
    
    <p class="text-menu">REALIZACAO PE</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link btn-logout" href="../cadastro.html">
    
    <p class="text-menu">SAIR</p>
  </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link" href="/pageNecessBiolog.html">
                
                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/pageNecessSocia.html">
                
                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" href="/pageNecessEspir.html">
                
                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
  <div class="fundo-botao">
    <a href="/necessidades/reliogisidadeespiritualidade.html" class="botao-padrao">
      <button>NECESSIDADES DE RELIGIOSIDADE E ESPIRITUALIDADE</button>
    </a>
  </div>

```



```
        </a>
    </div>
</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>
```

```
</html>
```

pages/pagesNecessidades/pageNecessSocia.html

```
<!DOCTYPE html>
<html lang="pt-br">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <title>HAsupport</title>
</head>
```

```
<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">
                <div class="container-fluid">
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <div>
                            <ul class="navbar-nav">
                                <li class="nav-item">
                                    <a class="nav-link" href="../../adm_area.html">
                                        
                                        <p class="text-menu">MENU INICIAL</p>
                                    </a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link" aria-current="page" href="../../menu1.html">
                                        
                                        <p class="text-menu">GUIA PE</p>
                                    </a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link active" aria-current="page" href="../../menu2.html">
                                        
                                        <p class="text-menu">REALIZACAO PE</p>
                                    </a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link btn-logout" href="../../cadastro.html">
```

```

        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link" href="/pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/pageNecessSocia.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="/pageNecessEspir.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </nav>
</div>
<div>
    <div class="fundo-botao">
        <a href="/necessidades/amoreaceitacao.html" class="botao-padrao">
            <button>NECESSIDADES DE AMOR E ACEITAÇÃO</button>
        </a>
        <a href="/necessidades/liberdadeeparticipacao.html" class="botao-padrao">
            <button>NECESSIDADES DE LIBERDADE E PARTICIPAÇÃO</button>
        </a>
        <a href="/necessidades/educacaosaude.html" class="botao-padrao">
            <button>NECESSIDADES DE EDUCAÇÃO PARA SAÚDE E APRENDIZAGEM</button>
        </a>
    </div>
</div>

```

```

</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/amoreaceitacao.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZAÇÃO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="../../cadastro.html">

```

```

        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="../../pageNecessSocia.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessEspir.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>AMOR E ACEITAÇÃO</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">
                <div>
                    <p>Diagnósticos/Resultados de enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rh-risc-contracep" type="checkbox" name="Aceitação do estado de saúde">
                    <label for="rv-risc-contracep">Aceitação do estado de saúde</label>
                </div>
            </form>
        </div>
    </div>

```

```

<div>
  <p>Intervenções de Enfermagem</p>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Estabelecer vínculo com o(a) usuário(a)
com hipertensão arterial">
  <label for="rh-ori-out-cont">Estabelecer vínculo com o(a) usuário(a) com hipertensão
arterial</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-sob-hor" type="checkbox" name="Incentivar a continuar com o hábito de
vida saudável">
  <label for="rh-ori-sob-hor">Incentivar a continuar com o hábito de vida saudável</label>
</div>
<div class="style-checkbox">
  <input id="rh-reg-inf" type="checkbox" name="Informar sobre as atividades de promoção da
saúde e prevenção de doenças promovidas
pela Unidade de Saúde">
  <label for="rh-reg-inf">Informar sobre as atividades de promoção da saúde e prevenção de
doenças promovidas
pela Unidade de Saúde</label>
</div>
<div class="style-checkbox">
  <input id="rh-susp-uso-cont" type="checkbox" name="Parabenizar o(a) usuário(a) com
hipertensão arterial pela aceitação da modificação do
estado de saúde">
  <label for="rh-susp-uso-cont">Parabenizar o(a) usuário(a) com hipertensão arterial pela
aceitação da modificação do
estado de saúde</label>
</div>
<div>
  <p id="retorno-amoraceitacao"></p>
</div>
<div>
  <button type="submit" id="btn-amoraceitacao-salvar">Salvar</button>
</div>
</form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/amoraceitacao.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

```

</html>

```

pages/pagesNecessidades/necessidades/atividadefisica.html

```

<!DOCTYPE html>
<html lang="pt-br">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">

```

```

<link rel="stylesheet" href="../../css/global.css">
<link rel="stylesheet" href="../../css/necessidades.css">
<title>HAsupport</title>
</head>

<body>
<div class="fundo1">
<div class="menu">
<nav class="navbar navbar-expand-lg">
<div class="container-fluid">
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<div>
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="../../adm_area.html">

<p class="text-menu">MENU INICIAL</p>
</a>
</li>
<li class="nav-item">
<a class="nav-link" aria-current="page" href="../../menu1.html">

<p class="text-menu">GUIA PE</p>
</a>
</li>
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="../../menu2.html">

<p class="text-menu">REALIZACAO PE</p>
</a>
</li>
<li class="nav-item">
<a class="nav-link btn-logout" href="../../cadastro.html">

<p class="text-menu">SAIR</p>
</a>
</li>
</ul>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
<nav class="navbar navbar-expand-lg">
<div class="container-fluid">
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>

```

```

<div class="collapse navbar-collapse" id="navbarNav">
  <div>
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" href="../../pageNecessBiolog.html">
          
          <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">
          
          <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="../../pageNecessEspir.html">
          
          <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
        </a>
      </li>
    </ul>
  </div>
</div>
</nav>
</div>
<div class="fundo-botao">
  <h3>ATIVIDADE FÍSICA</h3>
  <div class="form-necessidades">
    <form class="form-style" onsubmit="return false">
      <div>
        <p>Diagnósticos/Resultados de enfermagem</p>
      </div>
      <div class="style-checkbox">
        <input id="af-ad-reg" type="checkbox" name="Adesão ao regime exercício">
        <label for="af-ad-reg">Adesão ao regime exercício</label>
      </div>
      <div class="style-checkbox">
        <input id="af-fal-reg" type="checkbox" name="Falta de capacidade para gerir o regime de
exercício">
        <label for="af-fal-reg">Falta de capacidade para gerir o regime de exercício</label>
      </div>
      <div class="style-checkbox">
        <input id="af-fal-ade" type="checkbox" name="Falta de adesão ao exercício">
        <label for="af-fal-ade">Falta de adesão ao exercício</label>
      </div>
      <div class="style-checkbox">
        <input id="af-ris-tra" type="checkbox" name="Risco de trauma (Queda)">
        <label for="af-ris-tra">Risco de trauma (Queda)</label>
      </div>
      <div class="style-checkbox">
        <input id="af-tra-que" type="checkbox" name="Trauma (Queda)">
        <label for="af-tra-que">Trauma (Queda)</label>
      </div>
    </form>
  </div>

```

<p>Intervenções de Enfermagem</p>  
</div>  
<div class="style-checkbox">  
<input id="af-acomp-ev" type="checkbox" name="Acompanhar a evolução dos valores: peso, altura, pressão arterial, IMC, circunferência abdominal, relação cintura quadril e índice tornozelo-braquial, em todas as consultas, conforme descrito no exame físico">  
<label for="af-acomp-ev">Acompanhar a evolução dos valores: peso, altura, pressão arterial, IMC, circunferência abdominal, relação cintura quadril e índice tornozelo-braquial, em todas as consultas, conforme descrito no exame físico</label>  
</div>  
<div class="style-checkbox">  
<input id="af-afe-exer" type="checkbox" name="Aferir padrão de exercício">  
<label for="af-afe-exer">Aferir padrão de exercício</label>  
</div>  
<div class="style-checkbox">  
<input id="af-reg-exer" type="checkbox" name="Avaliar e registrar a adesão ao exercício a cada consulta de enfermagem">  
<label for="af-reg-exer">Avaliar e registrar a adesão ao exercício a cada consulta de enfermagem</label>  
</div>  
<div class="style-checkbox">  
<input id="af-reg-trat" type="checkbox" name="Avaliar e registrar o tratamento não medicamentoso no prontuário">  
<label for="af-reg-trat">Avaliar e registrar o tratamento não medicamentoso no prontuário</label>  
</div>  
<div class="style-checkbox">  
<input id="af-elog-reg-exer" type="checkbox" name="Elogiar a execução do regime de exercício">  
<label for="af-elog-reg-exer">Elogiar a execução do regime de exercício</label>  
</div>  
<div class="style-checkbox">  
<input id="af-exp-ef-exer" type="checkbox" name="Explicar sobre os efeitos do exercício físico na hipertensão arterial Incentivar a prática de exercícios regulares">  
<label for="af-exp-ef-exer">Explicar sobre os efeitos do exercício físico na hipertensão arterial Incentivar a prática de exercícios regulares</label>  
</div>  
<div class="style-checkbox">  
<input id="af-inc-con-pratic" type="checkbox" name="Incentivar e convidar o(a) usuário(a) com hipertensão arterial para participar de atividades de práticas saudáveis desenvolvidas na Unidade de Saúde">  
<label for="af-inc-con-pratic">Incentivar e convidar o(a) usuário(a) com hipertensão arterial para participar de atividades de práticas saudáveis desenvolvidas na Unidade de Saúde</label>  
</div>  
<div class="style-checkbox">  
<input id="af-inc-prat-ori" type="checkbox" name="Incentivar a prática de atividade física sob orientação">  
<label for="af-inc-prat-ori">Incentivar a prática de atividade física sob orientação</label>  
</div>  
<div class="style-checkbox">  
<input id="af-inc-plan-reab" type="checkbox" name="Incentivar a seguir plano terapêutico de reabilitação elaborado pela equipe">



<label for="af-inc-plan-reab">Incentivar a seguir plano terapêutico de reabilitação elaborado pela equipe</label>

</div>

<div class="style-checkbox">

<input id="af-inc-uso-loc" type="checkbox" name="Incentivar o uso de recursos que facilitam a locomoção">

<label for="af-inc-uso-loc">Incentivar o uso de recursos que facilitam a locomoção</label>

</div>

<div class="style-checkbox">

<input id="af-inc-usu-reab" type="checkbox" name="Incluir o(a) usuário(a) com hipertensão arterial em programas de reabilitação">

<label for="af-inc-usu-reab">Incluir o(a) usuário(a) com hipertensão arterial em programas de reabilitação</label>

</div>

<div class="style-checkbox">

<input id="af-inf-ap-sau" type="checkbox" name="Informar sobre os aparelhos sociais de práticas saudáveis de vida, se disponíveis">

<label for="af-inc-ap-sau">Informar sobre os aparelhos sociais de práticas saudáveis de vida, se disponíveis</label>

</div>

<div class="style-checkbox">

<input id="af-ori-usu-adpt" type="checkbox" name="Orientar o(a) usuário(a) com hipertensão arterial e os familiares sobre adaptações no ambiente domiciliar">

<label for="af-ori-usu-adpt">Orientar o(a) usuário(a) com hipertensão arterial e os familiares sobre adaptações no ambiente domiciliar</label>

</div>

<div class="style-checkbox">

<input id="af-ori-usu-med" type="checkbox" name="Orientar o(a) usuário(a) com hipertensão arterial e os familiares sobre medidas preventivas de traumas e adaptações no ambiente domiciliar">

<label for="af-ori-usu-med">Orientar o(a) usuário(a) com hipertensão arterial e os familiares sobre medidas preventivas de traumas e adaptações no ambiente domiciliar</label>

</div>

<div class="style-checkbox">

<input id="af-ori-nec-cam" type="checkbox" name="Orientar quanto à necessidade de fazer atividades físicas, como caminhadas">

<label for="af-ori-nec-cam">Orientar quanto à necessidade de fazer atividades físicas, como caminhadas</label>

</div>

<div class="style-checkbox">

<input id="af-ori-ef-pres" type="checkbox" name="Orientar quanto aos efeitos do exercício físico na redução do nível da pressão arterial">

<label for="af-ori-ef-pres">Orientar quanto aos efeitos do exercício físico na redução do nível da pressão arterial</label>

</div>

<div class="style-checkbox">

<input id="af-ori-risc-trau" type="checkbox" name="Orientar o(a) usuário(a) sobre riscos de trauma">

<label for="af-ori-risc-trau">Orientar o(a) usuário(a) sobre riscos de trauma</label>

</div>

<div>

<p id="retorno-atividadefisica"></p>

</div>

```
</div>
  <button type="submit" id="btn-atividadefisica-salvar">Salvar</button>
</div>
</form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/atividadefisica.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>
```

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="funcao1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li>
                </li>
                <li class="nav-item">
```

```

        <a class="nav-link active" aria-current="page" href="../../menu2.html">
            
            <p class="text-menu">REALIZAÇÃO PE</p>
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link btn-logout" href="../../cadastro.html">
            
            <p class="text-menu">SAIR</p>
        </a>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLÓGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="../../pageNecessSocia.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessEspir.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>EDUCAÇÃO PARA A SAÚDE</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">

```

```

<div>
  <p>Diagnósticos/Resultados de enfermagem</p>
</div>
<div class="style-checkbox">
  <input id="rh-risc-contracep" type="checkbox" name="Comportamento de busca de saúde">
  <label for="rv-risc-contracep">Comportamento de busca de saúde</label>
</div>
<div class="style-checkbox">
  <input id="rh-risc-contracep" type="checkbox" name="Conhecimento sobre a hipertensão
arterial adequado">
  <label for="rv-risc-contracep">Conhecimento sobre a hipertensão arterial adequado</label>
</div>
<div class="style-checkbox">
  <input id="rh-risc-contracep" type="checkbox" name="Falta de conhecimento sobre a
hipertensão arterial">
  <label for="rv-risc-contracep">Falta de conhecimento sobre a hipertensão arterial</label>
</div>
<div class="style-checkbox">
  <input id="rh-risc-contracep" type="checkbox" name="Habilidade para desempenhar a
manutenção da saúde">
  <label for="rv-risc-contracep">Habilidade para desempenhar a manutenção da saúde</label>
</div>
<div class="style-checkbox">
  <input id="rh-risc-contracep" type="checkbox" name="Manutenção da Saúde prejudicada">
  <label for="rv-risc-contracep">Manutenção da Saúde prejudicada</label>
</div>
<div>
  <p>Intervenções de Enfermagem</p>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Agendar retorno à Unidade de Saúde">
  <label for="rh-ori-out-cont">Agendar retorno à Unidade de Saúde</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Estabelecer relação de vínculo com o(a)
usuário(a) com hipertensão arterial">
  <label for="rh-ori-out-cont">Estabelecer relação de vínculo com o(a) usuário(a) com
hipertensão arterial</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Explicar o funcionamento da Unidade de
Saúde">
  <label for="rh-ori-out-cont">Explicar o funcionamento da Unidade de Saúde</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Incentivar a continuar com o hábito de
vida saudável">
  <label for="rh-ori-out-cont">Incentivar a continuar com o hábito de vida saudável</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Informar sobre os serviços disponíveis na
comunidade">
  <label for="rh-ori-out-cont">Informar sobre os serviços disponíveis na comunidade</label>
</div>
<div class="style-checkbox">
  <input id="rh-ori-out-cont" type="checkbox" name="Informar sobre as atividades de
promoção da saúde e prevenção de doenças promovidas

```

pela Unidade de Saúde">  
 <label for="rh-ori-out-cont">Informar sobre as atividades de promoção da saúde e prevenção de doenças promovidas  
 pela Unidade de Saúde</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Informar sobre os serviços disponíveis na Unidade de Saúde sobre hipertensão">  
 <label for="rh-ori-out-cont">Informar sobre os serviços disponíveis na Unidade de Saúde sobre hipertensão</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Instruir para procurar a Unidade de Saúde na presença de sinais e sintomas que  
 sinalizam níveis elevados de pressão arterial">  
 <label for="rh-ori-out-cont">Instruir para procurar a Unidade de Saúde na presença de sinais e sintomas que  
 sinalizam níveis elevados de pressão arterial</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar estratégias para a manutenção da saúde (prevenção de doenças, promoção e  
 manutenção da saúde)">  
 <label for="rh-ori-out-cont">Orientar estratégias para a manutenção da saúde (prevenção de doenças, promoção e  
 manutenção da saúde)</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar na visita domiciliar os familiares sobre o cuidado">  
 <label for="rh-ori-out-cont">Orientar na visita domiciliar os familiares sobre o cuidado</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar o(a) usuário(a) com hipertensão arterial e os familiares quanto ao  
 fluxograma da unidade">  
 <label for="rh-ori-out-cont">Orientar o(a) usuário(a) com hipertensão arterial e os familiares quanto ao  
 fluxograma da unidade</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar sobre as atividades desenvolvidas na Unidade de Saúde">  
 <label for="rh-ori-out-cont">Orientar sobre as atividades desenvolvidas na Unidade de Saúde</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar o(a) usuário(a) com hipertensão arterial e familiar sobre a hipertensão">  
 <label for="rh-ori-out-cont">Orientar o(a) usuário(a) com hipertensão arterial e familiar sobre a hipertensão</label>  
 </div>  
 <div class="style-checkbox">  
 <input id="rh-ori-out-cont" type="checkbox" name="Orientar sobre o processo de adoecimento">  
 <label for="rh-ori-out-cont">Orientar sobre o processo de adoecimento</label>

```

    </div>
    <div class="style-checkbox">
      <input id="rh-ori-out-cont" type="checkbox" name="Parabenizar o(a) usuário(a) com
hipertensão arterial pela aceitação da modificação do
      estado de saúde">
      <label for="rh-ori-out-cont">Parabenizar o(a) usuário(a) com hipertensão arterial pela
aceitação da modificação do
      estado de saúde</label>
    </div>
    <div class="style-checkbox">
      <input id="rh-ori-out-cont" type="checkbox" name="Pesquisar e registrar dados nas consultas
subsequentes sobre o conhecimento da
      hipertensão">
      <label for="rh-ori-out-cont">Pesquisar e registrar dados nas consultas subsequentes sobre o
conhecimento da
      hipertensão</label>
    </div>
    <div class="style-checkbox">
      <input id="rh-ori-out-cont" type="checkbox" name="Promover ações educativas de promoção
à saúde">
      <label for="rh-ori-out-cont">Promover ações educativas de promoção à saúde</label>
    </div>

    <div>
      <p id="retorno-educacao"></p>
    </div>
    <div>
      <button type="submit" id="btn-educacao-salvar">Salvar</button>
    </div>
  </form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/educacao.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/liberdadeeparticipacao.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

```

```

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="../../adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="../../menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="../../menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="../../cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">

```

```

<a class="nav-link" href="../pageNecessBiolog.html">
  
  <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
</a>
</li>
<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="../pageNecessSocia.html">
    
    <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="../pageNecessEspir.html">
    
    <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
  </a>
</li>
</div>
</ul>
</div>
</div>
</nav>
</div>
<div>
  <div class="fundo-botao">
    <h3>LIBERDADE E PARTICIPAÇÃO</h3>
    <div class="form-necessidades">
      <form class="form-style" onsubmit="return false">
        <div>
          <p>Diagnósticos/Resultados de enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rh-risc-contracep" type="checkbox" name="Falta de apoio social">
          <label for="rv-risc-contracep">Falta de apoio social</label>
        </div>
        <div class="style-checkbox">
          <input id="rh-risc-contracep" type="checkbox" name="Falta de apoio familiar">
          <label for="rv-risc-contracep">Falta de apoio familiar</label>
        </div>
        <div>
          <p>Intervenções de Enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rh-ori-out-cont" type="checkbox" name="Encaminhar para terapia de grupo de
apoio (serviço de referência), se necessário">
          <label for="rh-ori-out-cont">Encaminhar para terapia de grupo de apoio (serviço de
referência), se necessário</label>
        </div>
        <div class="style-checkbox">
          <input id="rh-ori-sob-hor" type="checkbox" name="Estabelecer, junto à equipe, estratégias
que possam favorecer atividades de apoio social">
          <label for="rh-ori-sob-hor">Estabelecer, junto à equipe, estratégias que possam favorecer
atividades de apoio social</label>
        </div>
        <div class="style-checkbox">

```



```

        <input id="rh-reg-inf" type="checkbox" name="Informar sobre as ações sociais desenvolvidas
na comunidade">
        <label for="rh-reg-inf">Informar sobre as ações sociais desenvolvidas na comunidade</label>
    </div>
    <div class="style-checkbox">
        <input id="rh-susp-uso-cont" type="checkbox" name="Orientar ao familiar do(a) usuário(a)
com hipertensão arterial sobre a modificação do
        estilo de vida">
        <label for="rh-susp-uso-cont">Orientar ao familiar do(a) usuário(a) com hipertensão arterial
sobre a modificação do
        estilo de vida</label>
    </div>
    <div class="style-checkbox">
        <input id="rh-susp-uso-cont" type="checkbox" name="Orientar a participação nos grupos da
Unidade de Saúde
        Sensibilizar a família a acompanhar as modificações da alimentação">
        <label for="rh-susp-uso-cont">Orientar a participação nos grupos da Unidade de Saúde
        Sensibilizar a família a acompanhar as modificações da alimentação</label>
    </div>
    <div class="style-checkbox">
        <input id="rh-susp-uso-cont" type="checkbox" name="Sensibilizar a família a acompanhar os
exercícios físicos">
        <label for="rh-susp-uso-cont">Sensibilizar a família a acompanhar os exercícios físicos</label>
    </div>
    <div class="style-checkbox">
        <input id="rh-susp-uso-cont" type="checkbox" name="Orientar ao familiar do(a) usuário(a)
com hipertensão arterial sobre a modificação do
        estilo de vida">
        <label for="rh-susp-uso-cont">POrientar ao familiar do(a) usuário(a) com hipertensão arterial
sobre a modificação do
        estilo de vida</label>
    </div>
    <div>
        <p id="retorno-liberdade"></p>
    </div>
    <div>
        <button type="submit" id="btn-liberdade-salvar">Salvar</button>
    </div>
</form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/liberdade.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

```

</html>

```

pages/pagesNecessidades/necessidades/neurologica.html

```

<!DOCTYPE html>
<html lang="pt-br">*fazer*

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
<link rel="stylesheet" href="../../css/global.css">
<link rel="stylesheet" href="../../css/necessidades.css">
<title>HAsupport</title>
</head>

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="../../adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="../../menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="../../menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="../../cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">

```

```

    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <div>
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" href="../pageNecessBiolog.html">
          
          <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="../pageNecessSocia.html">
          
          <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="../pageNecessEspir.html">
          
          <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
        </a>
      </li>
    </ul>
  </div>
</nav>
</div>
<div>
  <div class="fundo-botao">
    <h3>REGULAÇÃO NEUROLÓGICA</h3>
    <div class="form-necessidades">
      <form class="form-style" onsubmit="return false">
        <div>
          <p>Diagnósticos/Resultados de enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rn-ans-grau" type="checkbox" name="Ansiedade (especificar grau)">
          <label for="rn-ans-grau">Ansiedade (especificar grau)</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-dep-grau" type="checkbox" name="Depressão (especificar o grau)">
          <label for="rn-dep-grau">Depressão (especificar o grau)</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-risc-dep" type="checkbox" name="Risco de depressão">
          <label for="rn-risc-dep">Risco de depressão</label>
        </div>
        <div>
          <p>Intervenções de Enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rn-av-alt-estvida" type="checkbox" name="Avaliar e recomendar as alterações do
estilo de vida">
          <label for="rn-av-alt-estvida">Avaliar e recomendar as alterações do estilo de vida</label>
        </div>
      </form>
    </div>
  </div>

```

```

<div class="style-checkbox">
  <input id="rn-enc-serv-ref" type="checkbox" name="Encaminhar para serviço de referência,
se necessário">
  <label for="rn-enc-serv-ref">Encaminhar para serviço de referência, se necessário</label>
</div>
<div class="style-checkbox">
  <input id="rn-enc-autcui" type="checkbox" name="Encorajar as ações de autocuidado">
  <label for="rn-enc-autcui">Encorajar as ações de autocuidado</label>
</div>
<div class="style-checkbox">
  <input id="rn-enc-com" type="checkbox" name="Encorajar a comunicação">
  <label for="rn-enc-com">Encorajar a comunicação</label>
</div>
<div class="style-checkbox">
  <input id="rn-esc-usu" type="checkbox" name="Escutar o(a) usuário(a) com hipertensão
arterial durante a consulta de enfermagem">
  <label for="rn-esc-usu">Escutar o(a) usuário(a) com hipertensão arterial durante a consulta
de enfermagem</label>
</div>
<div class="style-checkbox">
  <input id="rn-esc-usu-ang" type="checkbox" name="Escutar o(a) usuário(a) com hipertensão
arterial para compreender suas necessidades e
angústias">
  <label for="rn-esc-usu-ang">Escutar o(a) usuário(a) com hipertensão arterial para
compreender suas necessidades e
angústias</label>
</div>
<div class="style-checkbox">
  <input id="rn-est-com" type="checkbox" name="Estimular a comunicação de angústia com o
tratamento específico">
  <label for="rn-ent-com">Estimular a comunicação de angústia com o tratamento
específico</label>
</div>
<div class="style-checkbox">
  <input id="rn-inc-part-ativ" type="checkbox" name="Incentivar a participação nas
atividades">
  <label for="rn-inc-part-ativ">Incentivar a participação nas atividades</label>
</div>
<div class="style-checkbox">
  <input id="rn-of-esc-qual" type="checkbox" name="Ofertar escuta qualificada">
  <label for="rn-of-esc-qual">Ofertar escuta qualificada</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar ao familiar sobre os cuidados,
acompanhamento do(a) usuário(a) com
hipertensão arterial">
  <label for="rn-or-cuid">Orientar ao familiar sobre os cuidados, acompanhamento do(a)
usuário(a) com
hipertensão arterial</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-at-soc" type="checkbox" name="Orientar a participação das atividades
sociais da comunidade e grupos na Unidade de
Saúde">
  <label for="rn-or-at-soc">Orientar a participação das atividades sociais da comunidade e
grupos na Unidade de
Saúde</label>

```

```

    </div>
    <div class="style-checkbox">
      <input id="rn-or-ag-estre" type="checkbox" name="Orientar sobre a identificação agentes
estressores">
      <label for="rn-or-ag-estre">Orientar sobre a identificação agentes estressores</label>
    </div>
    <div class="style-checkbox">
      <input id="rn-or-ativ-liv" type="checkbox" name="Orientar sobre as atividades de lazer
disponíveis na comunidade">
      <label for="rn-or-ativ-liv">Orientar sobre as atividades de lazer disponíveis na
comunidade</label>
    </div>
    <div class="style-checkbox">
      <input id="rn-or-med" type="checkbox" name="Orientar sobre o uso de medicamento, se
necessário">
      <label for="rn-or-med">Orientar sobre o uso de medicamento, se necessário</label>
    </div>
    <div class="style-checkbox">
      <input id="rn-or-tec-rel" type="checkbox" name="Orientar sobre técnicas de relaxamento">
      <label for="rn-or-tec-rel">Orientar sobre técnicas de relaxamento</label>
    </div>
    <div>
      <p id="retorno-regulacaoneurologica"></p>
    </div>
    <div>
      <button type="submit" id="btn-regulacaoneurologica-salvar">Salvar</button>
    </div>
  </form>
</div>
</div>
<script src="../../js/saveStorageNhb/regulacaoneurologica.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/nutricao.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">

```

```

<div class="container-fluid">
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <div>
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="../../adm_area.html">
            
            <p class="text-menu">MENU INICIAL</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="../../menu1.html">
            
            <p class="text-menu">GUIA PE</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="../../menu2.html">
            
            <p class="text-menu">REALIZACAO PE</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link btn-logout" href="../../cadastro.html">
            
            <p class="text-menu">SAIR</p>
          </a>
        </li>
      </ul>
    </div>
  </div>
</div>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="../../pageNecessBiolog.html">
                
                <p class="text-menu">NECESSIDADES PSICOBIOLÓGICAS</p>
              </a>
            </li>

```

```

<li class="nav-item">
  <a class="nav-link" aria-current="page" href="../pageNecessSocia.html">
    
    <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="../pageNecessEspir.html">
    
    <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
  </a>
</li>
</div>
</ul>
</div>
</div>
</nav>
</div>
<div>
  <div class="fundo-botao">
    <h3>NUTRIÇÃO</h3>
    <div class="form-necessidades">
      <form class="form-style" onsubmit="return false">
        <div>
          <p>Diagnósticos/Resultados de enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="nt-ads-reg" type="checkbox" name="Adesão ao regime dietético">
          <label for="nt-ads-reg">Adesão ao regime dietético</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-capac-prep" type="checkbox" name="Capacidade para preparar alimentos
saúdáveis">
          <label for="nt-capac-prep">Capacidade para preparar alimentos saúdáveis</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-capac-preju" type="checkbox" name="Capacidade prejudicada para preparar
alimentos saúdáveis">
          <label for="nt-capac-preju">Capacidade prejudicada para preparar alimentos
saúdáveis</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-emagr-saud" type="checkbox" name="Emagrecimento saúdável">
          <label for="nt-emagr-saud">Emagrecimento saúdável</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-emagr" type="checkbox" name="Emagrecimento">
          <label for="nt-emagr">Emagrecimento</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-excess-peso" type="checkbox" name="Excesso de peso">
          <label for="nt-excess-peso">Excesso de peso</label>
        </div>
        <div class="style-checkbox">
          <input id="nt-falta-reg" type="checkbox" name="Falta de adesão ao regime dietético">
          <label for="nt-falta-reg">Falta de adesão ao regime dietético</label>
        </div>
      </form>
    </div>
  </div>

```

```

<div class="style-checkbox">
  <input id="nt-hiperg" type="checkbox" name="Hiperglicemia">
  <label for="nt-hiperg">Hiperglicemia</label>
</div>
<div class="style-checkbox">
  <input id="nt-hipogl" type="checkbox" name="Hipoglicemia">
  <label for="nt-hipogl">Hipoglicemia</label>
</div>
<div class="style-checkbox">
  <input id="nt-inges-defic" type="checkbox" name="Ingestão de alimentos deficitária">
  <label for="nt-inges-defic">Ingestão de alimentos deficitária</label>
</div>
<div class="style-checkbox">
  <input id="nt-inges-excess" type="checkbox" name="Ingestão de alimentos excessiva">
  <label for="nt-inges-excess">Ingestão de alimentos excessiva</label>
</div>
<div class="style-checkbox">
  <input id="nt-obesi" type="checkbox" name="Obesidade">
  <label for="nt-obesi">Obesidade</label>
</div>
<div class="style-checkbox">
  <input id="nt-peso-adeq" type="checkbox" name="Peso corporal adequado">
  <label for="nt-peso-adeq">Peso corporal adequado</label>
</div>
<div class="style-checkbox">
  <input id="nt-peso-dimin" type="checkbox" name="Peso corporal diminuído">
  <label for="nt-peso-dimin">Peso corporal diminuído</label>
</div>
<div class="style-checkbox">
  <input id="nt-risc-inges-excess" type="checkbox" name="Risco de ingestão de alimentos
excessiva">
  <label for="nt-risc-inges-excess">Risco de ingestão de alimentos excessiva</label>
</div>
<div class="style-checkbox">
  <input id="nt-risc-inges-elev" type="checkbox" name="Risco de ingestão nutricional
elevado">
  <label for="nt-risc-inges-elev">Risco de ingestão nutricional elevado</label>
</div>
<div class="style-checkbox">
  <input id="nt-sobrep" type="checkbox" name="Sobrepeso">
  <label for="nt-sobrep">Sobrepeso</label>
</div>
<div>
  <p>Intervenções de Enfermagem</p>
</div>
<div class="style-checkbox">
  <input id="nt-acomp-evol" type="checkbox" name="Acompanhar a evolução dos valores: peso,
altura, pressão arterial, IMC, circunferência
abdominal, relação cintura quadril e índice tornozelo-braquial, em todas as consultas,
conforme descrito no exame físico">
  <label for="nt-acomp-evol">Acompanhar a evolução dos valores: peso, altura, pressão arterial,
IMC,
circunferência
abdominal, relação cintura quadril e índice tornozelo-braquial, em todas as consultas,
conforme descrito no exame físico</label>
</div>
<div class="style-checkbox">

```



```
<input id="nt-adeq-alim" type="checkbox" name="Adequar os alimentos ricos em proteínas
e/ou minerais de acordo com as condições
financeira">
<label for="nt-adeq-alim">Adequar os alimentos ricos em proteínas e/ou minerais de acordo
com as condições
financeira</label>
</div>
<div class="style-checkbox">
<input id="nt-agend-retor" type="checkbox" name="Agendar retorno à Unidade de Saúde">
<label for="nt-agend-retor">Agendar retorno à Unidade de Saúde</label>
</div>
<div class="style-checkbox">
<input id="nt-aval-medic" type="checkbox" name="Avaliar medicação">
<label for="nt-aval-medic">Avaliar medicação</label>
</div>
<div class="style-checkbox">
<input id="nt-aval-glic" type="checkbox" name="Avaliar os exames de glicemia">
<label for="nt-aval-glic">Avaliar os exames de glicemia</label>
</div>
<div class="style-checkbox">
<input id="nt-desenc-alim-gord" type="checkbox" name="Desencorajar a ingestão de
alimentos ricos em gordura, refrigerantes, sucos artificiais,
doces, guloseimas, bem como a redução de açúcar e carboidratos">
<label for="nt-desenc-alim-gord">Desencorajar a ingestão de alimentos ricos em gordura,
refrigerantes,
sucos artificiais,
doces, guloseimas, bem como a redução de açúcar e carboidratos</label>
</div>
<div class="style-checkbox">
<input id="nt-elogi-user" type="checkbox" name="Elogiar o(a) usuário(a) com hipertensão
arterial no cumprimento do regime dietético">
<label for="nt-elogi-user">Elogiar o(a) usuário(a) com hipertensão arterial no cumprimento
do regime
dietético</label>
</div>
<div class="style-checkbox">
<input id="nt-encam-aval-bucal" type="checkbox" name="Encaminhar o(a) usuário(a) com
hipertensão arterial para avaliação bucal, se necessário
e disponível">
<label for="nt-encam-aval-bucal">Encaminhar o(a) usuário(a) com hipertensão arterial para
avaliação bucal,
se necessário
e disponível</label>
</div>
<div class="style-checkbox">
<input id="nt-encam-consul-med" type="checkbox" name="Encaminhar o(a) usuário(a) com
hipertensão arterial para consulta com o médico, se
necessário">
<label for="nt-encam-consul-med">Encaminhar o(a) usuário(a) com hipertensão arterial para
consulta com o
médico, se
necessário</label>
</div>
<div class="style-checkbox">
<input id="nt-encam-consul-trimes" type="checkbox" name="Encaminhar para consultas
trimestrais, com o médico da equipe para avaliação">
```

```

        <label for="nt-encam-consul-trimes">Encaminhar para consultas trimestrais, com o médico
da equipe para
        avaliação</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-encam-nutri" type="checkbox" name="Encaminhar para nutricionista, se
necessário e disponível">
        <label for="nt-encam-nutri">Encaminhar para nutricionista, se necessário e disponível</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-encam-consul-trimes-eccess-peso" type="checkbox" name="Encaminhar para
consultas trimestrais, com o médico da equipe os(as) usuários(as)
        com hipertensão que apresentam excesso de peso">
        <label for="nt-encam-consul-trimes-eccess-peso">Encaminhar para consultas trimestrais, com
o médico da
        equipe os(as) usuários(as)
        com hipertensão que apresentam excesso de peso</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-encam-consul-trimes-sobrepeso" type="checkbox" name="Encaminhar para
consultas trimestrais, com o médico da equipe os(as) usuários(as)
        com hipertensão que apresentam sobrepeso">
        <label for="nt-encam-consul-trimes-sobrepeso">Encaminhar para consultas trimestrais, com
o médico da
        equipe os(as) usuários(as)
        com hipertensão que apresentam sobrepeso</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-encor-regim" type="checkbox" name="Encorajar a adoção do regime
dietético">
        <label for="nt-encor-regim">Encorajar a adoção do regime dietético</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-encor-manut-peso" type="checkbox" name="Encorajar a manutenção do
peso">
        <label for="nt-encor-manut-peso">Encorajar a manutenção do peso</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-ensin-anali-nutri" type="checkbox" name="Ensinar o(a) usuário(a) com
hipertensão arterial e sua família sobre a importância de
        analisar os nutrientes contidos nos alimentos">
        <label for="nt-ensin-anali-nutri">Ensinar o(a) usuário(a) com hipertensão arterial e sua família
sobre a
        importância de
        analisar os nutrientes contidos nos alimentos</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-ensin-alim-saud" type="checkbox" name="Ensinar o(a) usuário(a) com
hipertensão arterial e sua família sobre alimentação
        saudável">
        <label for="nt-ensin-alim-saud">Ensinar o(a) usuário(a) com hipertensão arterial e sua família
sobre
        alimentação
        saudável</label>
    </div>
    <div class="style-checkbox">

```

```

        <input id="nt-ensin-aprov-alim" type="checkbox" name="Ensinar sobre o aproveitamento
total dos alimentos">
        <label for="nt-ensin-aprov-alim">Ensinar sobre o aproveitamento total dos alimentos</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-estim-inges-carbo" type="checkbox" name="Estimular a ingestão de
carboidratos, proteínas e minerais, conforme as condições do
paciente">
        <label for="nt-estim-inges-carbo">Estimular a ingestão de carboidratos, proteínas e minerais,
conforme as
        condições do
        paciente</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-forn-mat-educ" type="checkbox" name="Fornecer material educativo sobre
preparo de alimentos, se disponível">
        <label for="nt-forn-mat-educ">Fornecer material educativo sobre preparo de alimentos, se
        disponível</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-incen-prep-part" type="checkbox" name="Incentivar a preparadora de
alimentos a participar de atividades de apoio social sobre
culinária">
        <label for="nt-incen-prep-part">Incentivar a preparadora de alimentos a participar de
atividades de apoio
        social sobre
        culinária</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-incen-part-grup" type="checkbox" name="Incentivar a participação nas
atividades de grupo da Unidade de Saúde">
        <label for="nt-incen-part-grup">Incentivar a participação nas atividades de grupo da Unidade
de
        Saúde</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-incen-comun-plano" type="checkbox" name="Incentivar a continuidade do
plano terapêutico">
        <label for="nt-incen-comun-plano">Incentivar a continuidade do plano terapêutico</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-incen-reduc" type="checkbox" name="Incentivar a redução de 5 a 10% do peso
inicial em até seis meses">
        <label for="nt-incen-reduc">Incentivar a redução de 5 a 10% do peso inicial em até seis
meses</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-incen-part-ativ" type="checkbox" name="Incentivar o(a) usuário(a) com
hipertensão arterial e familiares a participar das
atividades de educação em saúde">
        <label for="nt-incen-part-ativ">Incentivar o(a) usuário(a) com hipertensão arterial e familiares
a
        participar das
        atividades de educação em saúde</label>
    </div>
    <div class="style-checkbox">

```

```

        <input id="nt-inclu-frutas" type="checkbox" name="Incluir de 3 a 5 porções de frutas e
legumes/verduras em abundância na dieta (OBS: se
        usuário(a) com Diabetes Mellitus, orientar frutas após ingestão de proteínas)">
        <label for="nt-inclu-frutas">Incluir de 3 a 5 porções de frutas e legumes/verduras em
abundância na dieta
        (OBS: se
        usuário(a) com Diabetes Mellitus, orientar frutas após ingestão de proteínas)</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-info-fator-exce" type="checkbox" name="Informar sobre os fatores de risco
associados à ingestão alimentar excessiva">
        <label for="nt-info-fator-exce">Informar sobre os fatores de risco associados à ingestão
alimentar
        excessiva</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-inves-caus-emagre" type="checkbox" name="Investigar a causa do
emagrecimento">
        <label for="nt-inves-caus-emagre">Investigar a causa do emagrecimento</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-inves-caus-hipog" type="checkbox" name="Investigar causas da hipoglicemia">
        <label for="nt-inves-caus-hipog">Investigar causas da hipoglicemia</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-inves-caus-per-peso" type="checkbox" name="Investigar causas de perda de
peso">
        <label for="nt-inves-caus-per-peso">Investigar causas de perda de peso</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-inves-tipos-alim" type="checkbox" name="Investigar tipos de alimentos
ingeridos na rotina">
        <label for="nt-inves-tipos-alim">Investigar tipos de alimentos ingeridos na rotina</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-monit-regime" type="checkbox" name="Monitorar o regime dietético durante a
consulta de enfermagem">
        <label for="nt-monit-regime">Monitorar o regime dietético durante a consulta de
enfermagem</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-orient-hige-verd" type="checkbox" name="Orientar a higienizar verduras,
legumes e frutas antes do preparo dos alimentos">
        <label for="nt-orient-hige-verd">Orientar a higienizar verduras, legumes e frutas antes do
preparo dos
        alimentos</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-orient-inges-verd" type="checkbox" name="Orientar a ingestão de verduras,
legumes e frutas (utilizando o GUIA ALIMENTAR
        PARA A POPULAÇÃO BRASILEIRA – link de acesso:
        https://bvsms.saude.gov.br/bvs/publicacoes/guia_alimentar_populacao_brasileira_2ed
        .pdf)">
        <label for="nt-orient-inges-verd">Orientar a ingestão de verduras, legumes e frutas
(utilizando o GUIA
        ALIMENTAR
        PARA A POPULAÇÃO BRASILEIRA – link de acesso:

```

```
https://bvsms.saude.gov.br/bvs/publicacoes/guia_alimentar_populacao_brasileira_2ed
.pdf )</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-levar-maos" type="checkbox" name="Orientar a lavar as mãos antes de
manusear alimentos">
  <label for="nt-orient-levar-maos">Orientar a lavar as mãos antes de manusear
alimentos</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-nao-temp" type="checkbox" name="Orientar a não utilização de
temperos prontos na preparação dos alimentos">
  <label for="nt-orient-nao-temp">Orientar a não utilização de temperos prontos na
preparação dos
alimentos</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-redc-sodio" type="checkbox" name="Orientar a redução de sódio na
preparação dos alimentos">
  <label for="nt-orient-redc-sodio">Orientar a redução de sódio na preparação dos
alimentos</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-retir-gord" type="checkbox" name="Orientar a retirada da gordura
aparente das carnes antes de cozinhá-las">
  <label for="nt-orient-retir-gord">Orientar a retirada da gordura aparente das carnes antes de
cozinhá-las</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-aprov-alim" type="checkbox" name="Orientar sobre o aproveitamento
total dos alimentos">
  <label for="nt-orient-aprov-alim">Orientar sobre o aproveitamento total dos
alimentos</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-modif-alim" type="checkbox" name="Orientar a modificação no estilo
alimentar">
  <label for="nt-orient-modif-alim">Orientar a modificação no estilo alimentar</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-med-reduc-peso" type="checkbox" name="Orientar sobre as medidas de
redução do peso">
  <label for="nt-orient-med-reduc-peso">Orientar sobre as medidas de redução do
peso</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-ativ-alim" type="checkbox" name="Orientar o(a) usuário(a) com
hipertensão arterial e familiar a participação de atividade
sobre alimentação saudável na Unidade de Saúde">
  <label for="nt-orient-ativ-alim">Orientar o(a) usuário(a) com hipertensão arterial e familiar a
participação de atividade
sobre alimentação saudável na Unidade de Saúde</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-inges-vitam" type="checkbox" name="Orientar a ingestão de vitaminas e
minerais">
  <label for="nt-orient-inges-vitam">Orientar a ingestão de vitaminas e minerais</label>
```

```
</div>
<div class="style-checkbox">
  <input id="nt-orient-benef-est" type="checkbox" name="Orientar sobre os benefícios da
mudança no estilo de vida">
  <label for="nt-orient-benef-est">Orientar sobre os benefícios da mudança no estilo de
vida</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-frac-alim" type="checkbox" name="Orientar quanto ao fracionamento da
alimentação">
  <label for="nt-orient-frac-alim">Orientar quanto ao fracionamento da alimentação</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-exec-fisc" type="checkbox" name="Orientar sobre a prática de exercício
físico">
  <label for="nt-orient-exec-fisc">Orientar sobre a prática de exercício físico</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-alim-excess" type="checkbox" name="Orientar sobre as complicações da
alimentação em excesso">
  <label for="nt-orient-alim-excess">Orientar sobre as complicações da alimentação em
excesso</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-comp-hipog" type="checkbox" name="Orientar sobre as complicações da
hipoglicemia">
  <label for="nt-orient-comp-hipog">Orientar sobre as complicações da hipoglicemia</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-comp-nao-balanc" type="checkbox" name="Orientar sobre as
complicações de uma ingestão nutricional não balanceada">
  <label for="nt-orient-comp-nao-balanc">Orientar sobre as complicações de uma ingestão
nutricional não
balanceada</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-inges-frit" type="checkbox" name="Orientar sobre os riscos da ingestão
de frituras para as doenças cardiovasculares">
  <label for="nt-orient-inges-frit">Orientar sobre os riscos da ingestão de frituras para as
doenças
cardiovasculares</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-sobrepeso" type="checkbox" name="Orientar a pessoa com sobrepeso e
obesidade a prevenir e controlar a hipertensão por
meio de dieta, exercícios">
  <label for="nt-orient-sobrepeso">Orientar a pessoa com sobrepeso e obesidade a prevenir e
controlar a
hipertensão por
meio de dieta, exercícios</label>
</div>
<div class="style-checkbox">
  <input id="nt-orient-acucar-carbo" type="checkbox" name="Orientar quanto a alimentação
pobre em açúcar e carboidratos">
  <label for="nt-orient-acucar-carbo">Orientar quanto a alimentação pobre em açúcar e
carboidratos</label>
</div>
```

```
<div class="style-checkbox">
  <input id="nt-perg-prep-alim" type="checkbox" name="Perguntar e registrar sobre a
preparação de alimentos durante a consulta de enfermagem">
  <label for="nt-perg-prep-alim">Perguntar e registrar sobre a preparação de alimentos
durante a consulta de
  enfermagem</label>
</div>
<div class="style-checkbox">
  <input id="nt-pesq-comorb" type="checkbox" name="Pesquisar e registrar no prontuário e
ficha de acompanhamento sobre comorbidade">
  <label for="nt-pesq-comorb">Pesquisar e registrar no prontuário e ficha de acompanhamento
sobre
  comorbidade</label>
</div>
<div class="style-checkbox">
  <input id="nt-pesq-ult-hrs" type="checkbox" name="Pesquisar sobre a alimentação nas
últimas 24 horas">
  <label for="nt-pesq-ult-hrs">Pesquisar sobre a alimentação nas últimas 24 horas</label>
</div>
<div class="style-checkbox">
  <input id="nt-pesq-inges-nutri" type="checkbox" name="Pesquisar sobre a ingestão
nutricional">
  <label for="nt-pesq-inges-nutri">Pesquisar sobre a ingestão nutricional</label>
</div>
<div class="style-checkbox">
  <input id="nt-pesq-habit-alim" type="checkbox" name="Pesquisar sobre hábitos alimentares
que o(a) usuário(a) com hipertensão arterial tem
em casa.">
  <label for="nt-pesq-habit-alim">Pesquisar sobre hábitos alimentares que o(a) usuário(a) com
hipertensão
  arterial tem
  em casa.</label>
</div>
<div class="style-checkbox">
  <input id="nt-plan-reduc-peso" type="checkbox" name="Planejar conjuntamente as
estratégias de redução de peso">
  <label for="nt-plan-reduc-peso">Planejar conjuntamente as estratégias de redução de
peso</label>
</div>
<div class="style-checkbox">
  <input id="nt-recom-dimin-nutri" type="checkbox" name="Recomendar a diminuição de
nutrientes (gorduras, proteínas e carboidratos)">
  <label for="nt-recom-dimin-nutri">Recomendar a diminuição de nutrientes (gorduras,
proteínas e
  carboidratos)</label>
</div>
<div class="style-checkbox">
  <input id="nt-refor-higie-verd" type="checkbox" name="Reforçar a higienização das verduras
e legumes antes do preparo dos alimentos">
  <label for="nt-refor-higie-verd">Reforçar a higienização das verduras e legumes antes do
preparo dos
  alimentos</label>
</div>
<div class="style-checkbox">
  <input id="nt-refor-lavag-maos" type="checkbox" name="Reforçar a lavagem das mãos na
preparação dos alimentos">
```

```

        <label for="nt-refor-lavag-maos">Reforçar a lavagem das mãos na preparação dos
alimentos</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-refor-nao-temp" type="checkbox" name="Reforçar a não utilização de
temperos prontos na preparação dos alimentos">
        <label for="nt-refor-nao-temp">Reforçar a não utilização de temperos prontos na preparação
dos
        alimentos</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-refor-retir-gord" type="checkbox" name="Reforçar a preparadora de alimentos
para retirar a gordura aparente de carnes antes do
preparo">
        <label for="nt-refor-retir-gord">Reforçar a preparadora de alimentos para retirar a gordura
aparente de
        carnes antes do
        preparo</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-solic-exames-lab" type="checkbox" name="Solicitar exames laboratoriais para
avaliação (hemograma, glicemia de jejum,
        colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina sérica,
        potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),
        conforme protocolo">
        <label for="nt-solic-exames-lab">Solicitar exames laboratoriais para avaliação (hemograma,
glicemia de
        jejum,
        colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina sérica,
        potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),
        conforme protocolo</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-verif-diab" type="checkbox" name="Verificar se o(a) usuário(a) com hipertensão
arterial tem diagnóstico de diabetes
mellitus">
        <label for="nt-verif-diab">Verificar se o(a) usuário(a) com hipertensão arterial tem
diagnóstico de
        diabetes
        mellitus</label>
    </div>
    <div class="style-checkbox">
        <input id="nt-verif-hipog" type="checkbox" name="Verificar a dosagem dos hipoglicemiantes,
se possui diagnóstico de diabetes mellitus">
        <label for="nt-verif-hipog">Verificar a dosagem dos hipoglicemiantes, se possui diagnóstico
de diabetes
        mellitus</label>
    </div>
    <div>
        <p id="retorno-nutricao"></p>
    </div>
    <div>
        <button type="submit" id="btn-nutricao-salvar">Salvar</button>
    </div>
</form>
</div>
</div>

```



```
</div>
<script src="../../js/saveStorageNhb/nutricao.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>
```

```
</html>
```

pages/pagesNecessidades/necessidades/oxigenacao.html

```
<!DOCTYPE html>
<html lang="pt-br">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>
```

```
<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="../../cadastro.html">
```

```

        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link active" href="../../pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessEspir.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>OXIGENAÇÃO</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">
                <div>
                    <p>Diagnósticos/Resultados de enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="ox-resp-adeq" type="checkbox" name="Padrão respiratório adequado">
                    <label for="ox-resp-adeq">Padrão respiratório adequado</label>
                </div>
            </form>
        </div>
    </div>

```

```

<div class="style-checkbox">
  <input id="ox-resp-prej" type="checkbox" name="Padrão respiratório prejudicado">
  <label for="ox-resp-prej">Padrão respiratório prejudicado</label>
</div>
<div>
  <p>Intervenções de Enfermagem</p>
</div>
<div class="style-checkbox">
  <input id="ox-agen-retor" type="checkbox" name="Agendar retorno do(a) usuário(a) com hipertensão arterial à Unidade de Saúde, conforme avaliação e necessidade">
  <label for="ox-agen-retor">Agendar retorno do(a) usuário(a) com hipertensão arterial à Unidade de Saúde, conforme avaliação e necessidade</label>
</div>
<div class="style-checkbox">
  <input id="ox-encam-cons" type="checkbox" name="Encaminhar para consulta médica, se exame físico alterado">
  <label for="ox-encam-cons">Encaminhar para consulta médica, se exame físico alterado</label>
</div>
<div class="style-checkbox">
  <input id="ox-ensin-tecn" type="checkbox" name="Ensinar técnica de respiração lenta (10 respirações a cada minuto, durante 15 minutos, uma vez ao dia)">
  <label for="ox-ensin-tecn">Ensinar técnica de respiração lenta (10 respirações a cada minuto, durante 15 minutos, uma vez ao dia)</label>
</div>
<div class="style-checkbox">
  <input id="ox-parar-fumar" type="checkbox" name="Incentivar a parar de fumar">
  <label for="ox-parar-fumar">Incentivar a parar de fumar</label>
</div>
<div class="style-checkbox">
  <input id="ox-instr-encor" type="checkbox" name="Instruir e encorajar sobre o exercício respiratório (diafragmático) e a tosse eficaz">
  <label for="ox-instr-encor">Instruir e encorajar sobre o exercício respiratório (diafragmático) e a tosse eficaz</label>
</div>
<div class="style-checkbox">
  <input id="ox-inves-doenc" type="checkbox" name="Investigar sobre doenças respiratórias">
  <label for="ox-inves-doenc">Investigar sobre doenças respiratórias</label>
</div>
<div class="style-checkbox">
  <input id="ox-orien-habit" type="checkbox" name="Orientar a manter os hábitos de respiração">
  <label for="ox-orien-habit">Orientar a manter os hábitos de respiração</label>
</div>
<div class="style-checkbox">
  <input id="ox-orien-tabaco" type="checkbox" name="Orientar as complicações do tabaco para o aparelho respiratório">
  <label for="ox-orien-tabaco">Orientar as complicações do tabaco para o aparelho respiratório</label>
</div>
<div class="style-checkbox">

```

```

        <input id="ox-orien-respir" type="checkbox" name="Orientar como respirar durante as
atividades (inspirar pelo nariz e expirar pela
        boca)">
        <label for="ox-orien-respir">Orientar como respirar durante as atividades (inspirar pelo nariz
e expirar
        pela
        boca)</label>
    </div>
    <div class="style-checkbox">
        <input id="ox-pesq-alter" type="checkbox" name="Pesquisar sobre alterações na respiração
durante a consulta de enfermagem">
        <label for="ox-pesq-alter">Pesquisar sobre alterações na respiração durante a consulta de
enfermagem</label>
    </div>
    <div class="style-checkbox">
        <input id="ox-real-exam" type="checkbox" name="Realizar exame do aparelho respiratório
no(a) usuário(a) com
        hipertensão">
        <label for="ox-real-exam">Realizar exame do aparelho respiratório no(a) usuário(a) com
        hipertensão</label>
    </div>
    <div class="style-checkbox">
        <input id="ox-refor-ensin" type="checkbox" name="Reforçar o ensino de exercícios de
respiração profunda">
        <label for="ox-refor-ensin">Reforçar o ensino de exercícios de respiração profunda</label>
    </div>
    <div class="style-checkbox">
        <input id="ox-verif-vitais" type="checkbox" name="Verificar e registrar sinais vitais">
        <label for="ox-verif-vitais">Verificar e registrar sinais vitais</label>
    </div>
    <div>
        <p id="retorno-oxigenacao"></p>
    </div>
    <div>
        <button type="submit" id="btn-oxigenacao-salvar">Salvar</button>
    </div>
</form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/oxigenacao.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/verificaNivelUserFooter.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/regulacaohormonal.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
<link rel="stylesheet" href="../../css/global.css">
<link rel="stylesheet" href="../../css/necessidades.css">
<title>HAsupport</title>
</head>

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="../../adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="../../menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="../../menu2.html">
                  
                  <p class="text-menu">REALIZAÇÃO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="../../cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>

```

```

</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <div>
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" href="../pageNecessBiolog.html">
          
          <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="../pageNecessSocia.html">
          
          <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="../pageNecessEspir.html">
          
          <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
        </a>
      </li>
    </ul>
  </div>
</nav>
</div>
<div>
  <div class="fundo-botao">
    <h3>REGULAÇÃO HORMONAL</h3>
    <div class="form-necessidades">
      <form class="form-style" onsubmit="return false">
        <div>
          <p>Diagnósticos/Resultados de enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rh-risc-contracep" type="checkbox" name="Risco de complicações com o uso de
contraceptivos">
          <label for="rh-risc-contracep">Risco de complicações com o uso de contraceptivos</label>
        </div>
        <div>
          <p>Intervenções de Enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rh-ori-out-cont" type="checkbox" name="Orientar a usuária com hipertensão
arterial sobre outros métodos contraceptivos">
          <label for="rh-ori-out-cont">Orientar a usuária com hipertensão arterial sobre outros
métodos
contraceptivos</label>
        </div>
        <div class="style-checkbox">
          <input id="rh-ori-sob-hor" type="checkbox" name="Orientar a mulher sobre o uso do
contraceptivo no mesmo horário">
          <label for="rh-ori-sob-hor">Orientar a mulher sobre o uso do contraceptivo no mesmo
horário</label>
        </div>
      </form>
    </div>
  </div>

```

```

        <div class="style-checkbox">
            <input id="rh-reg-inf" type="checkbox" name="Registrar as informações da consulta no
prontuário">
            <label for="rh-reg-inf">Registrar as informações da consulta no prontuário</label>
        </div>
        <div class="style-checkbox">
            <input id="rh-susp-uso-cont" type="checkbox" name="Suspender o uso de contraceptivo, se
necessário">
            <label for="rh-susp-uso-cont">Suspender o uso de contraceptivo, se necessário</label>
        </div>
        <div>
            <p id="retorno-regulacao hormonal"></p>
        </div>
        <div>
            <button type="submit" id="btn-regulacao hormonal-salvar">Salvar</button>
        </div>
    </form>
</div>
</div>
<script src="../../js/saveStorageNhb/regulacao hormonal.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/regulacao vascular.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="../../css/necessidades.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">
                <div class="container-fluid">
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <div>
                            <ul class="navbar-nav">
                                <li class="nav-item">
                                    <a class="nav-link" href="../../adm_area.html">

```

```

        
        <p class="text-menu">MENU INICIAL</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" aria-current="page" href="../../menu1.html">
        
        <p class="text-menu">GUIA PE</p>
    </a>
</li>
</li>
<li class="nav-item">
    <a class="nav-link" aria-current="page" href="../../menu2.html">
        
        <p class="text-menu">REALIZAÇÃO PE</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link btn-logout" href="../../cadastro.html">
        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link active" href="../../pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLÓGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="../../pag eNecessEspir.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
                            </a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </nav>
</div>

```



```

        </li>
    </div>
</ul>
</div>
</div>
</nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>REGULAÇÃO VASCULAR</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">
                <div>
                    <p>Diagnósticos/Resultados de enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Controle de sintomas da
hipertensão arterial efetivo">
                    <label for="rv-risc-contracep">Controle de sintomas da hipertensão arterial
efetivo</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Controle de sintomas da
hipertensão arterial ineficaz">
                    <label for="rv-risc-contracep">Controle de sintomas da hipertensão arterial
ineficaz</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Diabetes">
                    <label for="rv-risc-contracep">Diabetes</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Edema Periférico">
                    <label for="rv-risc-contracep">Edema Periférico</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Edema Periférico melhorado">
                    <label for="rv-risc-contracep">Edema Periférico melhorado</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Epistaxe">
                    <label for="rv-risc-contracep">Epistaxe</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Frequência cardíaca alterada">
                    <label for="rv-risc-contracep">Frequência cardíaca alterada</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Frequência cardíaca
diminuída">
                    <label for="rv-risc-contracep">Frequência cardíaca diminuída</label>
                </div>
                <div class="style-checkbox">
                    <input id="rv-risc-contracep" type="checkbox" name="Hipertensão Arterial">
                    <label for="rv-risc-contracep">Hipertensão Arterial</label>
                </div>
                <div class="style-checkbox">

```

```

        <input id="rv-risc-contracep" type="checkbox" name="Hipotensão">
        <label for="rv-risc-contracep">Hipotensão</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Pressão Arterial baixa">
        <label for="rv-risc-contracep">Pressão Arterial baixa</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Pressão Arterial elevada">
        <label for="rv-risc-contracep">Pressão Arterial elevada</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Pressão Arterial normal">
        <label for="rv-risc-contracep">Pressão Arterial normal</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Pulso radial palpável">
        <label for="rv-risc-contracep">Pulso radial palpável</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Risco de Edema Periférico">
        <label for="rv-risc-contracep">Risco de Edema Periférico</label>
    </div>

    <p>Intervenções de Enfermagem</p>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Agendar retorno à Unidade de
Saúde para controle do estado de saúde, conforme
avaliação e necessidade">
        <label for="rv-risc-contracep">Agendar retorno à Unidade de Saúde para controle do
estado de
saúde, conforme
avaliação e necessidade</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Anotar sinais vitais em
caderneta, cartão individual e prontuário do(a) usuário(a) com
hipertensão arterial">
        <label for="rv-risc-contracep">Anotar sinais vitais em caderneta, cartão individual e
prontuário do(a) usuário(a) com
hipertensão arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Avaliar a necessidade de uso
de medicação de urgência">
        <label for="rv-risc-contracep">Avaliar a necessidade de uso de medicação de
urgência</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Avaliar e registrar histórico da
frequência cardíaca no prontuário">
        <label for="rv-risc-contracep">Avaliar e registrar histórico da frequência cardíaca no
prontuário</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Avaliar e registrar na consulta
de enfermagem a presença de edema periférico">

```

```
<label for="rv-risc-contracep">Avaliar e registrar na consulta de enfermagem a
presença de
    edema periférico</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Avaliar o tratamento utilizado
no controle dos sintomas">
    <label for="rv-risc-contracep">Avaliar o tratamento utilizado no controle dos
sintomas</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Avaliar se o(a) usuário(a)
possui diagnóstico de hipertensão arterial">
    <label for="rv-risc-contracep">Avaliar se o(a) usuário(a) possui diagnóstico de
hipertensão
    arterial</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Elogiar sobre os valores
normais da pressão arterial (menor ou igual a 120/80mmHg)">
    <label for="rv-risc-contracep">Elogiar sobre os valores normais da pressão arterial
(menor
    ou igual a 120/80mmHg)</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Encaminhar ao médico para
avaliação da medicação, se necessário">
    <label for="rv-risc-contracep">Encaminhar ao médico para avaliação da medicação, se
necessário</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Encaminhar para serviço de
referência o(a) usuário(a) com hipertensão arterial
apresentando urgência cardiológica">
    <label for="rv-risc-contracep">Encaminhar para serviço de referência o(a) usuário(a)
com
    hipertensão arterial
    apresentando urgência cardiológica</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Encorajar a manutenção de
uma alimentação saudável">
    <label for="rv-risc-contracep">Encorajar a manutenção de uma alimentação
saudável</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Explicar a ação do Ácido
Acetilsalicílico">
    <label for="rv-risc-contracep">Explicar a ação do Ácido Acetilsalicílico</label>
</div>
<div class="style-checkbox">
    <input id="rv-risc-contracep" type="checkbox" name="Incentivar a continuidade dos
hábitos de vida">
    <label for="rv-risc-contracep">Incentivar a continuidade dos hábitos de vida</label>
</div>
<div class="style-checkbox">
```

```

        <input id="rv-risc-contracep" type="checkbox" name="Incentivar o(a) usuário(a) com
hipertensão arterial e doenças associadas a participar de
atividades de grupo na Unidade de Saúde">
        <label for="rv-risc-contracep">Incentivar o(a) usuário(a) com hipertensão arterial e
doenças
        associadas a participar de
        atividades de grupo na Unidade de Saúde</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Informar sobre os riscos do
edema para a hipertensão arterial">
        <label for="rv-risc-contracep">Informar sobre os riscos do edema para a hipertensão
arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Medir a frequência cardíaca">
        <label for="rv-risc-contracep">Medir a frequência cardíaca</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Monitorar a glicemia, se
necessário">
        <label for="rv-risc-contracep">Monitorar a glicemia, se necessário</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Monitorar e registrar a pressão
arterial no mínimo três vezes na semana em dias
alternados, se necessário">
        <label for="rv-risc-contracep">Monitorar e registrar a pressão arterial no mínimo três
vezes
        na semana em dias
        alternados, se necessário</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar para hábitos de
alimentação saudável">
        <label for="rv-risc-contracep">Orientar para hábitos de alimentação saudável</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar o(a) usuário(a) com
hipertensão arterial a buscar a Unidade de Saúde na
presença de sinais e sintomas de elevação da pressão arterial">
        <label for="rv-risc-contracep">Orientar o(a) usuário(a) com hipertensão arterial a
buscar a
        Unidade de Saúde na
        presença de sinais e sintomas de elevação da pressão arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar o(a) usuário(a) com
hipertensão arterial a buscar a Unidade de Saúde na
presença de sinais e sintomas de elevação da pressão arterial e alteração do nível
glicêmico">
        <label for="rv-risc-contracep">Orientar o(a) usuário(a) com hipertensão arterial a
buscar a
        Unidade de Saúde na
        presença de sinais e sintomas de elevação da pressão arterial e alteração do nível
glicêmico</label>
    </div>

```

```
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar a posologia da
medicação">
    <label for="rv-risc-contracep">Orientar a posologia da medicação</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar para não tomar
medicação sem prescrição, quando surgir outras doenças">
    <label for="rv-risc-contracep">Orientar para não tomar medicação sem prescrição,
quando
    surgir outras doenças</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar quanto ao controle da
pressão arterial">
    <label for="rv-risc-contracep">Orientar quanto ao controle da pressão arterial</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre a importância e
necessidade de uma alimentação hipossódica adequada">
    <label for="rv-risc-contracep">Orientar sobre a importância e necessidade de uma
alimentação
    hipossódica adequada</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre a posologia da
medicação">
    <label for="rv-risc-contracep">Orientar sobre a posologia da medicação</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre a prevenção do
pé diabético, se diagnóstico de diabetes mellitus">
    <label for="rv-risc-contracep">Orientar sobre a prevenção do pé diabético, se
diagnóstico de
    diabetes mellitus</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre as complicações
da diabetes para hipertensão arterial">
    <label for="rv-risc-contracep">Orientar sobre as complicações da diabetes para
hipertensão
    arterial</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre o tratamento da
hipertensão e da diabetes">
    <label for="rv-risc-contracep">Orientar sobre o tratamento da hipertensão e da
diabetes</label>
</div>
<div class="style-checkbox">
  <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre o tratamento
medicamento e não medicamentoso">
    <label for="rv-risc-contracep">Orientar sobre o tratamento medicamento e não
medicamentoso</label>
</div>
<div class="style-checkbox">
```

```

        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre os benefícios da
atividade física">
        <label for="rv-risc-contracep">Orientar sobre os benefícios da atividade física</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre os benefícios da
pressão arterial controlada">
        <label for="rv-risc-contracep">Orientar sobre os benefícios da pressão arterial
controlada</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre os sinais e
sintomas da frequência cardíaca alterada">
        <label for="rv-risc-contracep">Orientar sobre os sinais e sintomas da frequência
cardíaca
        alterada</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre os sinais e
sintomas da frequência cardíaca diminuída">
        <label for="rv-risc-contracep">Orientar sobre os sinais e sintomas da frequência
cardíaca
        diminuída</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre os valores
normais da pressão arterial">
        <label for="rv-risc-contracep">Orientar sobre os valores normais da pressão
arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre prevenção de
edema periférico">
        <label for="rv-risc-contracep">Orientar sobre prevenção de edema periférico</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre repouso">
        <label for="rv-risc-contracep">Orientar sobre repouso</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Orientar sobre complicações
da hipertensão arterial">
        <label for="rv-risc-contracep">Orientar sobre complicações da hipertensão
arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Pedir que descreva os sintomas
pregressos e atuais">
        <label for="rv-risc-contracep">Pedir que descreva os sintomas pregressos e
atuais</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Perguntar ao(a) usuário(a) qual
a sua pressão arterial usual">
        <label for="rv-risc-contracep">Perguntar ao(a) usuário(a) qual a sua pressão arterial
usual</label>
    </div>

```

```

        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar a alimentação
pregressa
            Pesquisar e orientar sobre o uso de medicamento para controle dos sintomas">
                <label for="rv-risc-contracep">Pesquisar a alimentação pregressa
                    Pesquisar e orientar sobre o uso de medicamento para controle dos
sintomas</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar e registrar no
prontuário os relatos da melhora do edema periférico">
                <label for="rv-risc-contracep">Pesquisar e registrar no prontuário os relatos da
melhora do
                    edema periférico</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar se o(a) usuário(a) faz
uso de medicação que contem Ácido Acetilsalicílico">
                <label for="rv-risc-contracep">Pesquisar se o(a) usuário(a) faz uso de medicação que
contem
                    Ácido Acetilsalicílico</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar sintomatologia
pregressa">
                <label for="rv-risc-contracep">Pesquisar sintomatologia pregressa</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar sobre a prática de
atividade física excessiva">
                <label for="rv-risc-contracep">Pesquisar sobre a prática de atividade física
excessiva</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar sobre alimentação">
                <label for="rv-risc-contracep">Pesquisar sobre alimentação</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar sobre o uso de
medicação">
                <label for="rv-risc-contracep">Pesquisar sobre o uso de medicação</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar sobre o uso de
medicamentos para hipertensão">
                <label for="rv-risc-contracep">Pesquisar sobre o uso de medicamentos para
hipertensão</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Pesquisar uso de
superdosagem da medicação">
                <label for="rv-risc-contracep">Pesquisar uso de superdosagem da medicação</label>
        </div>
        <div class="style-checkbox">
            <input id="rv-risc-contracep" type="checkbox" name="Reavaliar e esclarecer a
conduta de enfermagem utilizada">
                <label for="rv-risc-contracep">Reavaliar e esclarecer a conduta de enfermagem

```

```
        utilizada</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Recomendar que em casos de
sintomas moderados e intensos procurar um serviço de
saúde">
        <label for="rv-risc-contracep">Recomendar que em casos de sintomas moderados e
intensos
        procurar um serviço de
saúde</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Registrar no prontuário as
informações completas da consulta">
        <label for="rv-risc-contracep">Registrar no prontuário as informações completas da
consulta</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Solicitar exames laboratoriais
para avaliação (hemograma, glicemia de jejum,
colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina sérica,
potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),
conforme protocolo">
        <label for="rv-risc-contracep">Solicitar exames laboratoriais para avaliação
(hemograma,
        glicemia de jejum,
        colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina
        sérica,
        potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),
        conforme protocolo</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar e registrar a pressão
arterial por três vezes na semana, se necessário">
        <label for="rv-risc-contracep">Verificar e registrar a pressão arterial por três vezes na
semana, se necessário</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar e registrar a pressão
arterial conforme o procedimento já descrito">
        <label for="rv-risc-contracep">Verificar e registrar a pressão arterial conforme o
procedimento já descrito</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar pressão arterial de
paciente com AVC no domicílio">
        <label for="rv-risc-contracep">Verificar pressão arterial de paciente com AVC no
domicílio</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar pressão arterial de
pacientes em crise hipertensiva">
        <label for="rv-risc-contracep">Verificar pressão arterial de pacientes em crise
hipertensiva</label>
    </div>
    <div class="style-checkbox">
```



```

        <input id="rv-risc-contracep" type="checkbox" name="Verificar pressão arterial em
todas as consultas do(a) usuário(a) com hipertensão
        arterial">
        <label for="rv-risc-contracep">Verificar pressão arterial em todas as consultas do(a)
        usuário(a) com hipertensão
        arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar se o(a) usuário(a)
possui diagnóstico de hipertensão arterial ">
        <label for="rv-risc-contracep">Verificar se o(a) usuário(a) possui diagnóstico de
        hipertensão arterial</label>
    </div>
    <div class="style-checkbox">
        <input id="rv-risc-contracep" type="checkbox" name="Verificar se houve algum
agente externo que interferiu nos valores da pressão arterial">
        <label for="rv-risc-contracep">Verificar se houve algum agente externo que interferiu
nos
        valores da pressão arterial</label>
    </div>
    <div>
        <p id="retorno-regulacaovascular"></p>
    </div>
    <div>
        <button type="submit" id="btn-regulacaovascular-salvar">Salvar</button>
    </div>
</form>
</div>
</div>
<script src="../../js/saveStorageNhb/regulacaovascular.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

```
</html>
```

pages/pagesNecessidades/necessidades/reliogisidadeespiritualidade.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="../../css/necessidades.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">
                <div class="container-fluid">

```

```

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <div>
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="../../adm_area.html">
                        
                        <p class="text-menu">MENU INICIAL</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" aria-current="page" href="../../menu1.html">
                        
                        <p class="text-menu">GUIA PE</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page" href="../../menu2.html">
                        
                        <p class="text-menu">REALIZAÇÃO PE</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link btn-logout" href="../../cadastro.html">
                        
                    </a>
                </li>
            </ul>
        </div>
    </div>
</div>
</div>
<div class="sub-menu-necess">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <div>
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link" href="../../pageNecessBiolog.html">
                                
                                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
                            </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">

```

```

        
        <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link active" href="../../pageNecessEspir.html">
        
        <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
    </a>
</li>
</div>
</ul>
</div>
</div>
</nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>RELIGIOSIDADE/ESPIRITUALIDADE</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">
                <div>
                    <p>Diagnósticos/Resultados de enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rh-risc-contracep" type="checkbox" name="Crença de saúde prejudicada">
                    <label for="rv-risc-contracep">Crença de saúde prejudicada</label>
                </div>
                <div>
                    <p>Intervenções de Enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rh-ori-out-cont" type="checkbox" name="Compartilhar experiências exitosas de
enfrentamento do estado de saúde">
                    <label for="rh-ori-out-cont">Compartilhar experiências exitosas de enfrentamento do estado
de saúde</label>
                </div>
                <div class="style-checkbox">
                    <input id="rh-ori-sob-hor" type="checkbox" name="Escutar e registrar as dificuldades
relatadas">
                    <label for="rh-ori-sob-hor">Escutar e registrar as dificuldades relatadas</label>
                </div>
                <div class="style-checkbox">
                    <input id="rh-reg-inf" type="checkbox" name="Fortalecer vínculo de confiança">
                    <label for="rh-reg-inf">Fortalecer vínculo de confiança</label>
                </div>
                <div class="style-checkbox">
                    <input id="rh-susp-uso-cont" type="checkbox" name="Informar sobre as atividades de
promoção da saúde e prevenção de doenças promovidas
pela Unidade de Saúde">
                    <label for="rh-susp-uso-cont">Informar sobre as atividades de promoção da saúde e
prevenção de doenças promovidas
pela Unidade de Saúde</label>
                </div>
                <div class="style-checkbox">
                    <input id="rh-reg-inf" type="checkbox" name="Promover medidas de melhorias da saúde">
                    <label for="rh-reg-inf">Promover medidas de melhorias da saúde</label>

```

```

    </div>
    <div>
      <p id="retorno-religiosidade"></p>
    </div>
    <div>
      <button type="submit" id="btn-religiosidade-salvar">Salvar</button>
    </div>
  </form>
</div>
</div>
<script src="../../js/saveStorageNhb/religiosidade.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/repousoesono.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>

```

```

</li>
</li>
<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="../../menu2.html">
    
    <p class="text-menu">REALIZAÇÃO PE</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link btn-logout" href="../../cadastro.html">
    
    <p class="text-menu">SAIR</p>
  </a>
</li>
</ul>
</div>
</div>
</div>
</nav>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="../../pageNecessBiolog.html">
                
                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">
                
                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="../../pageNecessEspir.html">
                
                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </div>
  </nav>
</div>
<div>
  <div class="fundo-botao">

```

```

<h3>SONO E REPOUSO</h3>
<div class="form-necessidades">
  <form class="form-style" onsubmit="return false">
    <div>
      <p>Diagnósticos/Resultados de enfermagem</p>
    </div>
    <div class="style-checkbox">
      <input id="sr-apn" type="checkbox" name="Apneia">
      <label for="sr-apn">Apneia</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-sono-adeq" type="checkbox" name="Sono adequado">
      <label for="sr-sa">Capacidade para preparar alimentos saudáveis</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-sono-preju" type="checkbox" name="Sono prejudicado">
      <label for="sr-sono-preju">Sono prejudicado</label>
    </div>
    <div>
      <p>Intervenções de Enfermagem</p>
    </div>
    <div class="style-checkbox">
      <input id="sr-disc-med" type="checkbox" name="Discutir com o(a) usuário(a) com hipertensão arterial e familiar sobre medidas de conforto">
      <label for="sr-disc-med">Discutir com o(a) usuário(a) com hipertensão arterial e familiar sobre medidas de conforto</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-enc-rel" type="checkbox" name="Encorajar ao relaxamento">
      <label for="sr-enc-rel">Encorajar ao relaxamento</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-ens-tec" type="checkbox" name="Ensinar técnicas de relaxamento">
      <label for="sr-ens-tec">Ensinar técnicas de relaxamento</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-est-al" type="checkbox" name="Estimular atividade que promova um estado de alerta adequado durante o dia">
      <label for="sr-est-al">Estimular atividade que promova um estado de alerta adequado durante o dia</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-inf-serv" type="checkbox" name="Informar sobre os serviços de terapia comunitária da Unidade de Saúde">
      <label for="sr-inf-serv">Informar sobre os serviços de terapia comunitária da Unidade de Saúde</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-or-rui" type="checkbox" name="Orientar sobre o ambiente livre de ruídos">
      <label for="sr-or-rui">Orientar sobre o ambiente livre de ruídos</label>
    </div>
    <div class="style-checkbox">
      <input id="sr-or-rel-mus" type="checkbox" name="Orientar sobre técnicas de relaxamento muscular">
      <label for="sr-or-rel-mus">Orientar sobre técnicas de relaxamento muscular</label>
    </div>
  </form>
</div>

```

```
<div class="style-checkbox">
  <input id="sr-or-pra-fis" type="checkbox" name="Orientar quanto à prática da atividade física">
  <label for="sr-or-pra-fis">Orientar quanto à prática da atividade física</label>
</div>
<div class="style-checkbox">
  <input id="sr-imp-desc" type="checkbox" name="Reforçar a importância de um descanso satisfatório para a recuperação da saúde">
  <label for="sr-imp-desc">Reforçar a importância de um descanso satisfatório para a recuperação da saúde</label>
</div>
<div>
  <p id="retorno-sonorepouso"></p>
</div>
<div>
  <button type="submit" id="btn-sonoerepouso-salvar">Salvar</button>
</div>
</form>
</div>
</div>
</div>
<script src="../../js/saveStorageNhb/sonorepouso.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/necessidades.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
```

```

<a class="nav-link" href="../../adm_area.html">
  
  <p class="text-menu">MENU INICIAL</p>
</a>
</li>
<li class="nav-item">
  <a class="nav-link" aria-current="page" href="../../menu1.html">
    
    <p class="text-menu">GUIA PE</p>
  </a>
</li>
</li>
<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="../../menu2.html">
    
    <p class="text-menu">REALIZACAO PE</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link btn-logout" href="../../cadastro.html">
    
    <p class="text-menu">SAIR</p>
  </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
navigation="navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="../../pageNecessBiolog.html">
                
                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="../../pageNecessSocia.html">
                
                <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
              </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="../../pageNecessEspir.html">
                
                <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </div>
  </nav>
</div>

```



```

        </a>
    </li>
</div>
</ul>
</div>
</div>
</nav>
</div>
<div>
    <div class="fundo-botao">
        <h3>PERCEPÇÃO DOS ÓRGÃOS DOS SENTIDOS</h3>
        <div class="form-necessidades">
            <form class="form-style" onsubmit="return false">
                <div>
                    <p>Diagnósticos/Resultados de enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rn-ans-grau" type="checkbox" name="Controle da dor efetivo">
                    <label for="rn-ans-grau">Controle da dor efetivo</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-dep-grau" type="checkbox" name="Controle da dor ineficaz">
                    <label for="rn-dep-grau">Controle da dor ineficaz</label>
                </div>
                <div>
                    <p>Intervenções de Enfermagem</p>
                </div>
                <div class="style-checkbox">
                    <input id="rn-av-alt-estvida" type="checkbox" name="Agendar retorno à Unidade de Saúde">
                    <label for="rn-av-alt-estvida">Agendar retorno à Unidade de Saúde</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-enc-serv-ref" type="checkbox" name="Ensinar técnicas de relaxamento">
                    <label for="rn-enc-serv-ref">Encaminhar para serviço de referência, se necessário</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-enc-autcui" type="checkbox" name="Identificar as características da dor
pregressa e atual, se existir">
                    <label for="rn-enc-autcui">Identificar as características da dor pregressa e atual, se
existir</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-enc-com" type="checkbox" name="Identificar local da dor">
                    <label for="rn-enc-com">Identificar local da dor</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-esc-usu" type="checkbox" name="Orientar quanto às complicações das dores
na hipertensão arterial">
                    <label for="rn-esc-usu">Orientar quanto às complicações das dores na hipertensão
arterial</label>
                </div>
                <div class="style-checkbox">
                    <input id="rn-esc-usu-ang" type="checkbox" name="Pesquisar e orientar sobre o uso de
medicamento">
                    <label for="rn-esc-usu-ang">Pesquisar e orientar sobre o uso de medicamento</label>
                </div>
            </form>
        </div>
    </div>

```

```

        <div class="style-checkbox">
            <input id="rn-est-com" type="checkbox" name="Pesquisar sobre a localização pregressa da
dor">
            <label for="rn-ent-com">Pesquisar sobre a localização pregressa da dor</label>
        </div>
        <div class="style-checkbox">
            <input id="rn-inc-part-ativ" type="checkbox" name="Recomendar a procurar um hospital em
casos de dor moderada e intensa">
            <label for="rn-inc-part-ativ">Recomendar a procurar um hospital em casos de dor moderada
e intensa</label>
        </div>
        <div class="style-checkbox">
            <input id="rn-of-esc-qual" type="checkbox" name="Registrar as informações da consulta no
prontuário">
            <label for="rn-of-esc-qual">Registrar as informações da consulta no prontuário</label>
        </div>
        <div class="style-checkbox">
            <input id="rn-or-cuid" type="checkbox" name="Verificar e registrar sinais vitais no
prontuário">
            <label for="rn-or-cuid">Verificar e registrar sinais vitais no prontuário</label>
        </div>
        <div>
            <p id="retorno-sentidos"></p>
        </div>
        <div>
            <button type="submit" id="btn-sentidos-salvar">Salvar</button>
        </div>
    </form>
</div>
</div>
<script src="../../js/saveStorageNhb/sentidos.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagesNecessidades/necessidades/terapeutica.html

```

<!DOCTYPE html>
<html lang="pt-br"> *fazer*

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="../../css/necessidades.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">

```

```

<div class="container-fluid">
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <div>
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="../../adm_area.html">
            
            <p class="text-menu">MENU INICIAL</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="../../menu1.html">
            
            <p class="text-menu">GUIA PE</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="../../menu2.html">
            
            <p class="text-menu">REALIZAÇÃO PE</p>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link btn-logout" href="../../cadastro.html">
            
            <p class="text-menu">SAIR</p>
          </a>
        </li>
      </ul>
    </div>
  </div>
</div>
</div>
<div class="sub-menu-necess">
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <div>
          <ul class="navbar-nav">
            <li class="nav-item">
              <a class="nav-link active" href="../../pageNecessBiolog.html">
                
                <p class="text-menu">NECESSIDADES PSICOBIOLOGICAS</p>
              </a>
            </li>

```

```

<li class="nav-item">
  <a class="nav-link" aria-current="page" href="../pageNecessSocia.html">
    
    <p class="text-menu">NECESSIDADES PSICOSOCIAIS</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="../pageNecessEspir.html">
    
    <p class="text-menu">NECESSIDADES PSICOESPIRITUAIS</p>
  </a>
</li>
</div>
</ul>
</div>
</div>
</nav>
</div>
<div>
  <div class="fundo-botao">
    <h3>TERAPÊUTICA</h3>
    <div class="form-necessidades">
      <form class="form-style" onsubmit="return false">
        <div>
          <p>Diagnósticos/Resultados de enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rn-ans-grau" type="checkbox" name="Adesão ao regime terapêutico">
          <label for="rn-ans-grau">Adesão ao regime terapêutico</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-dep-grau" type="checkbox" name="Adesão ao regime medicamentoso">
          <label for="rn-dep-grau">Adesão ao regime medicamentoso</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-dep-grau" type="checkbox" name="Falta de adesão ao regime terapêutico">
          <label for="rn-dep-grau">Falta de adesão ao regime terapêutico</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-dep-grau" type="checkbox" name="Falta de adesão ao tratamento
medicamentoso">
          <label for="rn-dep-grau">Falta de adesão ao tratamento medicamentoso</label>
        </div>
        <div>
          <p>Intervenções de Enfermagem</p>
        </div>
        <div class="style-checkbox">
          <input id="rn-av-alt-estvida" type="checkbox" name="Avaliar e registrar os resultados de
exames no prontuário">
          <label for="rn-av-alt-estvida">Avaliar e registrar os resultados de exames no
prontuário</label>
        </div>
        <div class="style-checkbox">
          <input id="rn-enc-serv-ref" type="checkbox" name="Elogiar a adesão ao regime
medicamentoso">
          <label for="rn-enc-serv-ref">Elogiar a adesão ao regime medicamentoso</label>
        </div>
      </form>
    </div>
  </div>

```

```
<div class="style-checkbox">
  <input id="rn-enc-autcui" type="checkbox" name="Encorajar a adesão ao regime
terapêutico">
  <label for="rn-enc-autcui">Encorajar a adesão ao regime terapêutico</label>
</div>
<div class="style-checkbox">
  <input id="rn-enc-com" type="checkbox" name="Encorajar a adesão do tratamento
medicamentoso de forma contínua">
  <label for="rn-enc-com">Encorajar a adesão do tratamento medicamentoso de forma
contínua</label>
</div>
<div class="style-checkbox">
  <input id="rn-esc-usu" type="checkbox" name="Estimar a adesão ao tratamento">
  <label for="rn-esc-usu">Estimar a adesão ao tratamento</label>
</div>
<div class="style-checkbox">
  <input id="rn-esc-usu-ang" type="checkbox" name="Estimar nas consultas subsequentes a
adesão ao regime terapêutico">
  <label for="rn-esc-usu-ang">Estimar nas consultas subsequentes a adesão ao regime
terapêutico</label>
</div>
<div class="style-checkbox">
  <input id="rn-est-com" type="checkbox" name="Explicar a ação do medicamento">
  <label for="rn-ent-com">Explicar a ação do medicamento</label>
</div>
<div class="style-checkbox">
  <input id="rn-inc-part-ativ" type="checkbox" name="Facilitar acesso ao tratamento">
  <label for="rn-inc-part-ativ">Facilitar acesso ao tratamento</label>
</div>
<div class="style-checkbox">
  <input id="rn-of-esc-qual" type="checkbox" name="Informar sobre os serviços oferecidos na
Unidade de Saúde">
  <label for="rn-of-esc-qual">Informar sobre os serviços oferecidos na Unidade de
Saúde</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Informar quais medicamentos anti-
hipertensivos são disponíveis nos serviços">
  <label for="rn-or-cuid">Informar quais medicamentos anti-hipertensivos são disponíveis nos
serviços</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar o retorno a Unidade de Saúde em
caso de efeito adverso">
  <label for="rn-or-cuid">Orientar o retorno a Unidade de Saúde em caso de efeito
adverso</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre a periodicidade do
acompanhamento">
  <label for="rn-or-cuid">Orientar sobre a periodicidade do acompanhamento</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre acondicionamento dos
medicamentos">
  <label for="rn-or-cuid">Orientar sobre acondicionamento dos medicamentos</label>
</div>
```

```
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre as complicações da
hipertensão arterial quando não segue o regime
  terapêutico">
  <label for="rn-or-cuid">Orientar sobre as complicações da hipertensão arterial quando não
segue o regime
  terapêutico</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre efeitos adversos">
  <label for="rn-or-cuid">Orientar sobre efeitos adversos</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre o regime terapêutico">
  <label for="rn-or-cuid">Orientar sobre o regime terapêutico</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre o uso e os horários dos
medicamentos">
  <label for="rn-or-cuid">Orientar sobre o uso e os horários dos medicamentos</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar como é realizado o diagnóstico da
hipertensão arterial">
  <label for="rn-or-cuid">Orientar como é realizado o diagnóstico da hipertensão
arterial</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre os benefícios do regime
terapêutico para a qualidade de vida">
  <label for="rn-or-cuid">Orientar sobre os benefícios do regime terapêutico para a qualidade
de vida</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre os benefícios do tratamento">
  <label for="rn-or-cuid">Orientar sobre os benefícios do tratamento</label>
</div>
<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Orientar sobre os direitos do(a) usuário(a)
com hipertensão arterial no acesso a
  medicação">
  <label for="rn-or-cuid">Orientar sobre os direitos do(a) usuário(a) com hipertensão arterial no
acesso a
  medicação</label>
</div>

<div class="style-checkbox">
  <input id="rn-or-cuid" type="checkbox" name="Solicitar exames laboratoriais para avaliação
(hemograma, glicemia de jejum,
  colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina sérica,
  potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),conforme
protocolo">
  <label for="rn-or-cuid">Solicitar exames laboratoriais para avaliação (hemograma, glicemia de
jejum,
  colesterol total, colesterol HDL, triglicerídeos, hemoglobina glicada, creatinina sérica,
  potássio, EQU, microalbumina na urina, albumina, ácido úrico, uréia, TGO e TGP),conforme
protocolo"</label>
```

```

    </div>
    <div class="style-checkbox">
      <input id="rn-or-cuid" type="checkbox" name="Verificar e registrar a pressão arterial durante
a consulta de enfermagem">
      <label for="rn-or-cuid">Verificar e registrar a pressão arterial durante a consulta de
enfermagem</label>
    </div>
    <div>
      <p id="retorno-terapeutica"></p>
    </div>
    <div>
      <button type="submit" id="btn-terapeutica-salvar">Salvar</button>
    </div>
  </form>
</div>
</div>
<script src="../../js/saveStorageNhb/terapeutica.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/ImpressaoReferencia.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.10.1/html2pdf.bundle.min.js"
integrity="sha512-
GsLlZN/3F2ErC5ifS5QtgpiJtWd43JWSulgh7mbzZ8zBps+dvLusV+eNQATqgA/HdeKFVgA5v3S/cIrLF7Qnlg
==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
</head>
<body>
  <div class="Content">
    <!-- EDITAR AQUI -->
    <h1>Título do texto</h1>

    <h2>Sub Título de exemplo para um vídeo no YouTube</h2>

    <p>Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos,
    e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma
    bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos.
    Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a
    editoração eletrônica, permanecendo essencialmente inalterado.
    Se popularizou na década de 60, quando a Letraset lançou decalques
    contendo passagens de Lorem Ipsum, e mais recentemente quando passou
    a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.
    </p>
    <!-- ATÉ AQUI -->
  </div>

```

```
<!-- BOTÃO DE DOWNLOAD -->
<button onclick="downloadPDF()" >Download</button>
```

```
<script src="script.js"></script>
</body>
</html>
```

pages/Menu\_intermediario.html

```
<!DOCTYPE html>
<html lang="pt-br">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/cadastro.css">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <title>HAsupport</title>
</head>
```

```
<body>
  <div class="fundo-cadastro-global">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="/cadastro.html">
```



```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/cadastro.css">
  <link rel="stylesheet" href="../css/displayNone.css">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

  <title>HAsupport</title>
</head>

<body>
  <div class="fundo-cadastro-global">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
```

```

</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <div>
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link active" href="/adm_area.html">
          
          <p class="text-menu">MENU INICIAL</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="/menu1.html">
          
          <p class="text-menu">GUIA PE</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="/menu2.html">
          
          <p class="text-menu">REALIZACAO PE</p>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link btn-logout" href="/cadastro.html">
          
          <p class="text-menu">SAIR</p>
        </a>
      </li>
    </ul>
  </div>
</div>
</div>
<div class="funcao-cadastro">
  <div class="form">
    <div id="enabled-case-admin">
      <a href="/alt_equipe.html">
        <button id="alterar">ALTERAR CADASTRO</button>
      </a>
      <a href="/busca_equipe.html">
        <button id="alterar">BUSCAR EQUIPE</button>
      </a>
    </div>
    <a href="/busca_user.html" class="disable-btn-busca">
      <button id="alterar">Buscar usuário(a)/relatório individual</button>
    </a>
    <a href="/cad_usuario.html" class="disable-btn-consulta">
      <button id="alterar">Cadastrar novo(a) usuário(a)/iniciar Processo de Enfermagem
(PE)</button>
    </a>
    <a href="/menu1.html" class="disable-btn-protocol">
      <button id="alterar">Guia para o Processo de Enfermagem (PE)</button>
    </a>
  </div>
</div>

```

```

    </a>
    <a href="/pagePrintRelatorio.html" class="disable-btn-relatorio">
      <button>Consolidados usuários(as)</button>
    </a>
  </div>
  <a class="btn-ajuda">
    <button id="btn-ajuda">
      <span>Ajuda</span>
    </button>
  </a>
  <p id="retorno-config-nivel-user"></p>
  <p id="retorno-ajuda-user"></p>
</div>
</div>
<script src="../js/enableAdmin.js"></script>
<script src="../js/logoutSistema.js"></script>
<script src="../js/ajudaUser.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/alt\_equipe.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/cadastro.css">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo-cadastro-global">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu"> MENU INICIAL </p>
                  </a>
                </li>
                <li class="nav-item">

```

```

        <a class="nav-link" aria-current="page" href="/menu1.html">
            
            <p class="text-menu">GUIA PE</p>
        </a>
    </li>
</li>
<li class="nav-item">
    <a class="nav-link" aria-current="page" href="/menu2.html">
        
        <p class="text-menu">REALIZACAO PE</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link btn-logout" href="/cadastro.html">
        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="fundo-cadastro">
    <div class="form" onsubmit="return false">
        <form>
            <div>
                <input id="login" class="login" name="login" required="required" type="text"
placeholder="Login da Equipe" />
            </div>
            <div>
                <input id="senha" class="senha" name="senha" required="required" type="password"
placeholder="Senha da Equipe" />
            </div>
            <div>
                <input id="membros" class="membros" name="membros" required="required" type="text"
placeholder="Membros da Equipe" />
            </div>
            <div>
                <button id="alterar">ALTERAR</button>
            </div>
            <div>
                <p id="retorno-up-equipe"></p>
            </div>
        </form>
    </div>
</div>
</div>
</div>
<script type="module" src="../backend/equipe_js/update.js"></script>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/busca\_equipe.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/cadastro.css">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo-cadastro-global">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="/cadastro.html">
                    
                    <p class="text-menu">SAIR</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>
</body>
</html>
```



```

        <a class="nav-link" href="/adm_area.html">
            
            <p class="text-menu">MENU INICIAL</p>
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" aria-current="page" href="/menu1.html">
            
            <p class="text-menu">GUIA PE</p>
        </a>
    </li>
    </li>
    <li class="nav-item">
        <a class="nav-link" aria-current="page" href="/menu2.html">
            
            <p class="text-menu">REALIZACAO PE</p>
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link btn-logout" href="/cadastro.html">
            
            <p class="text-menu">SAIR</p>
        </a>
    </li>
</ul>
</div>
</div>
</div>
</div>
</nav>
</div>
<div class="fundo-cadastro">
    <div class="form" onsubmit="return false">
        <form>
            <div>
                <input id="user_sus" class="user_sus" name="user_sus" required="required" type="text"
placeholder="Cartão SUS, CPF ou Nome" />
            </div>
            <div>
                <button id="buscar">BUSCAR</button>
            </div>
        </form>
    </div>
    <p id="retorno-busca-user"></p>
    <div id="result">
        <!--Lista de Usuários do SUS-->
    </div>
    <!--
    <div class="btn-cad-novo-user">
        <a href="/cad_usuario.html">
            <button>CADASTRAR NOVO USUÁRIO</button>
        </a>
    </div>
    -->
</div>
</div>
<script type="module" src="/backend/user_sus/busca.js"></script>
<script type="text/javascript" src="/backend/user_sus/seleciona.js"></script>

```

```

<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>

</body>

</html>

pages/cad_equipe.html

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/cadastro.css">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo-cadastro-global">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="/cadastro.html">
                    

```



```

        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</nav>
</div>
<div class="fundo-cadastro">
    <div class="form" onsubmit="return false">
        <form>
            <div>
                <input id="login" class="login" name="login" required="required" type="text"
placeholder="Login da Equipe" />
            </div>
            <div>
                <input id="senha" class="senha" name="senha" required="required" type="password"
placeholder="Senha da Equipe" />
            </div>
            <div>
                <input id="membros" class="membros" name="membros" required="required" type="text"
placeholder="Membros da Equipe" />
            </div>
            <div>
                <!--apos cadastro redirecionar para adm_area.html-->
                <button id="cadastrar">Cadastrar</button>
            </div>
            <div>
                <p id="retorno-cad-equipe">...</p>
            </div>
        </form>
    </div>
</div>
<div>
    <script type="module" src="../../backend/equipe_js/cadastro.js"></script>
    <script src="../../js/logoutSistema.js"></script>
    <script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/cad\_usuario.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="../../css/cadastro.css">

```

```

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
<title>HAsupport</title>
</head>

<body>
<div class="fundo-cadastro-global">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="/adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="/menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="/cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="fundo-cadastro">
    <div class="form" onsubmit="return false">
      <form>
        <div>
          <input id="nome" class="nome" name="nome" required="required" type="text"
placeholder="Nome" />
        </div>
      </div>
    </div>
  </div>

```

```

        <input id="dataNasc" class="data_nascimento" name="data_nascimento" required="required"
type="date" placeholder="Data de Nascimento" />
    </div>
    <div>
        <input id="cpf" class="cpf" name="cpf" required="required" type="number" maxlength="11"
placeholder="CPF" oninput="javascript: if (this.value.length > this.maxLength) this.value =
this.value.slice(0, this.maxLength);" />
    </div>
    <div>
        <input id="cartSus" class="cartao_sus" name="cartao_sus" required="required"
type="number" maxlength="15" placeholder="Cartão do Sus" oninput="javascript: if
(this.value.length > this.maxLength) this.value = this.value.slice(0, this.maxLength);" />
    </div>
    <div>
        <input id="agenteSus" class="agente_sus" name="agente_sus" required="required"
type="text" placeholder="Agente Comunitaria(o) de Saúde" />
    </div>
    <div class="input-radio">
        <div>
            <input id="masc" name="sexo" type="radio" value="Masculino" checked>
            <label for="masc">Masculino</label>
        </div>
        <div>
            <input id="femi" name="sexo" type="radio" value="Feminino">
            <label for="femi">Feminino</label>
        </div>
    </div>
    <div>
        <button id="cadastrar">CADASTRAR</button>
    </div>
    <div>
        <p id="retorno-cad-user-sucess">...</p>
    </div>
</form>
</div>
</div>
<script type="module" src="../../backend/user_sus/cadastro.js"></script>
<script type="module" src="../../js/saveCadastros/cadUsuario.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

</html>

pages/cadastro.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/cadastro.css">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <title>HAsupport</title>
</head>

```

```

<body>
  <div class="fundo-cadastro-global">
    <div class="fundo-cadastro">
      <!--

      <div>
        <a href="/adm_area.html">Area ADM (link temporario)</a>
      </div>

      -->
      <div class="img">
        
      </div>
      <div class="form" onsubmit="return false">
        <form>
          <div>
            <p id="acesso-negado"></p>
          </div>
          <div>
            <input id="login" class="login" name="login" required="required" type="text"
placeholder="Nome de usuário" />
          </div>
          <div>
            <input id="senha" class="senha" name="senha" required="required" type="password"
placeholder="Senha" />
          </div>
          <div>
            <!--apos login redirecionar para adm_area.html-->
            <button id="logar">ENTRAR</button>
          </div>
          <div>
            <a href="/cad_equipe.html">
              CADASTRAR-SE
            </a>
          </div>
        </form>
      </div>
    </div>
    <div class="footer">
      <div>
        <h4>Desenvolvido por<a href="/pageDireitos.html">Clique Aqui</a></h4>
      </div>
      <div class="footer-logos">
        
        
        
      </div>
    </div>
  </div>
  <script>
    localStorage.clear();
    console.log("Storage Limpo");
  </script>
  <script type="module" src="../backend/equipe_js/login.js"></script>
</body>

```

</html>

pages/menu1.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

<link rel="stylesheet" href="../css/global.css">

<title>HAsupport</title>

</head>

<body>

<div class="fundo1">

<div class="menu">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="/adm\_area.html">



<p class="text-menu">MENU INICIAL</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="/menu1.html">



<p class="text-menu">GUIA PE</p>

</a>

</li>

</li>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu2.html">



<p class="text-menu">REALIZACAO PE</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link btn-logout" href="/cadastro.html">



<p class="text-menu">SAIR</p>

</a>

</li>

</ul>

</div>

```

        </div>
    </div>
</nav>
</div>
<div class="fundo-botao">
    <a href="/pageHistoricoEnf.html" class="botao-padrao">
        <button>HISTÓRICO DE ENFERMAGEM</button>
    </a>
    <a href="/pageDiagnosEnfExemplo.html" class="botao-padrao">
        <button>DIAGNÓSTICO DE ENFERMAGEM</button>
    </a>
    <a href="/pagePlanejEnfer.html" class="botao-padrao">
        <button>PLANEJAMENTO DE ENFERMAGEM</button>
    </a>
    <a href="/pageImplement.html" class="botao-padrao">
        <button>IMPLEMENTAÇÃO</button>
    </a>
    <a href="/pageAvaEnf.html" class="botao-padrao">
        <button>AVALIAÇÃO DE ENFERMAGEM</button>
    </a>
</div>
</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/menu2.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">
                <div class="container-fluid">
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <div>
                            <ul class="navbar-nav">
                                <li class="nav-item">
                                    <a class="nav-link" href="/adm_area.html">

```

```

        
        <p class="text-menu">MENU INICIAL</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" aria-current="page" href="/menu1.html">
        
        <p class="text-menu">GUIA PE</p>
    </a>
</li>
</li>
<li class="nav-item">
    <a class="nav-link active" aria-current="page" href="/menu2.html">
        
        <p class="text-menu">REALIZACAO PE</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link btn-logout" href="/cadastro.html">
        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</nav>
</div>
<div class="fundo-botao">
    <a href="/pageExamesLab.html" class="botao-padrao">
        <button>EXAMES LABORATORIAIS</button>
    </a>
    <a href="/pageAvaClinic.html" class="botao-padrao">
        <button>AVALIAÇÃO CLÍNICA E CÁLCULO DO RISCO CARDIOVASCULAR</button>
    </a>
    <a href="/pageDiagnosEnf.html" class="botao-padrao">
        <button>DIAGNÓSTICO DE ENFERMAGEM</button>
    </a>
</div>
    <a class="botao-padrao">
        <button id="btn-finaliza-consulta">FINALIZAR CONSULTA</button>
    </a>
</div>
<div>
    <p id="retorno-finaliza-cosulta">...</p>
</div>
</div>
</div>
<script type="module" src="../../backend/user_sus/loadDataUser.js"></script>
<script type="module" src="../../backend/connection/atendimento_db.js"></script>
<script src="../../js/finalizaConsulta.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>
</html>

```

pages/pageAvaClinic.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/displayNone.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery.mask@1.14.16/jquery.mask.js"
integrity="sha256-yE5LLp5HSQ/z+hJeCqkz9hdjNkk1jaiGG0tDCraumnA="
crossorigin="anonymous"></script>
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu"> MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu1.html">
                    
                    <p class="text-menu"> GUIA PE</p>
                  </a>
                </li>
                <li>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="/menu2.html">
                    
                    <p class="text-menu"> REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="/cadastro.html">
                    
                    <p class="text-menu"> SAIR</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>
</body>
```



```

        </div>
    </div>
</div>
</nav>
</div>
<div class="fundo-botao">
    <a class="botao-padrao">
        <button id="btn-rastreamento">RASTREAMENTO DA HIPERTENSÃO ARTERIAL<span
class="icon-expand"></span></button>
    </a>
    <div class="text-none-rastreamento text-none-style">
        <div>
            <h4>Rastreamento da hipertensão arterial</h4>
            <div class="text-style-pure2">
                <p>Para o rastreamento recomenda-se a verificação na Unidade pelo menos duas vezes
na
                    semana, por duas semanas e se possível, conciliar com a Monitorização Residencial da
                    Pressão Arterial (MRPA), sendo duas vezes ao dia durante sete dias ou três vezes ao
dia
                    durante cinco dias (PROTOCOLO DE ENFERMAGEM: HIPERTENSÃO E
                    DIABETES, 2021).</p>
            </div>
        </div>
        <div>
            <h4>Valores aferidos</h4>
            <form class="form-style" onsubmit="return false">
                <div>
                    <input id="pas" type="number" placeholder="PAS: MMHG">
                </div>
                <div>
                    <input id="pad" type="number" placeholder="PAD: MMHG">
                </div>
                <div>
                    <button id="btn-calcular-pas-pad">CALCULAR</button>
                </div>
            </form>
            <p>Classificação: <span id="retorno-calculo-pas-pad">Aguardando valores...</span></p>
        </div>
    </div>
    <a class="botao-padrao">
        <button id="btn-antropom">DADOS ANTROPOMÉTRICOS<span class="icon-expand"></span></button>
    </a>
    <div class="text-none-antropom text-none-style">
        <div class="flex-ajuste">
            <a class="botao-padrao botao-padrao-interno">
                <button id="btn-imc">ÍNDICE DE MASSA CORPORAL (IMC)<span class="icon-
expand"></span></button>
            </a>
            <div class="text-none-imc text-none-style-interno">
                <h4>ÍNDICE DE MASSA CORPORAL (IMC)</h4>
                <div>
                    <div>
                        <p>IDADE:<span id="imc-retorno-idade">...</span>&emsp;SEXO:<span id="imc-
retorno-sexo">...</span></p>
                    </div>
                    <form action="" class="form-style-interno" onsubmit="return false">

```

```

        <div>
            <input id="imc-peso" type="text" placeholder="PESO(KG): 80">
        </div>
        <div>
            <input id="imc-altura" type="text" placeholder="ALTURA(M): 1.70">
        </div>
        <div>
            <button id="btn-imc-calcular">CALCULAR</button>
        </div>
    </form>
    <p>Classificação: <span id="retorno-calculo-imc">Aguardando valores...</span></p>
    <p>IMC: <span id="retorno-calculo-imc2">Aguardando valores...</span></p>

</div>
</div>
</div>
<div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
        <button id="btn-circ-abdom">CIRCUNFERÊNCIA ABDOMINAL<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-circ-abdom text-none-style-interno">
        <h4>CIRCUNFERÊNCIA ABDOMINAL</h4>
        <div>
            <div>
                <p>SEXO:<span id="circ-abdom-retorno-sexo">...</span></p>
            </div>
            <form action="" class="form-style-interno" onsubmit="return false">
                <div>
                    <input id="circ-abdom-cm" type="number" placeholder="(CM)">
                </div>
                <div>
                    <button id="btn-circ-abdom-calcular">CALCULAR</button>
                </div>
            </form>
            <p>Classificação: <span id="retorno-calculo-circ-abdom">Aguardando
valores...</span></p>
        </div>
    </div>
</div>
</div>
<div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
        <button id="btn-cint-quadr">RELAÇÃO CINTURA-QUADRIL<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-cint-quadr text-none-style-interno">
        <h4>RELAÇÃO CINTURA-QUADRIL</h4>
        <div>
            <div>
                <p>SEXO:<span id="cint-quadr-retorno-sexo">...</span></p>
            </div>
            <form action="" class="form-style-interno" onsubmit="return false">
                <div>
                    <input id="cint-quadr-abdom-cm" type="number" placeholder="(CM)
CIRCUNFERÊNCIA ABDOMINAL">
                </div>
                <div>

```

```

        <input id="cint-quadr-quadr-cm" type="number" placeholder="(CM)
CIRCUNFERÊNCIA DO QUADRIL">
    </div>
    <div>
        <button id="btn-cint-quadr-calculador">CALCULAR</button>
    </div>
</form>
<p>Classificação: <span id="retorno-calculo-cint-quadr">Aguardando
valores...</span></p>
</div>
</div>
</div>
<div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
        <button id="btn-torn-braq">ÍNDICE TORNOZELO-BRAQUIAL<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-torn-braq text-none-style-interno">
        <h4>ÍNDICE TORNOZELO-BRAQUIAL</h4>
        <div>
            <form action="" class="form-style-interno" onsubmit="return false">
                <div>
                    <input id="torn-braq-pas-torn" type="number" placeholder="VALOR DA MAIOR
PAS DO TORNOZELO">
                </div>
                <div>
                    <input id="torn-braq-pas-bra" type="number" placeholder="VALOR DA PAS DO
BRAÇO">
                </div>
                <div>
                    <button id="btn-torn-braq-calculador">CALCULAR</button>
                </div>
            </form>
            <p>Classificação: <span id="retorno-calculo-torn-braq">Aguardando
vlores...</span></p>
        </div>
    </div>
</div>
</div>
<a class="botao-padrao">
    <button id="btn-estrat-risco">ESTRATIFICAÇÃO DO RISCO CARDIOVASCULAR<span
class="icon-expand"></span></button>
</a>
<div class="text-none-estrat-risco text-none-style">
    <h4>ESTRATIFICAÇÃO DO RISCO CARDIOVASCULAR</h4>
    <div class="text-style-pure2">
        <p> Para o cálculo da Escala de Risco de Framingham (ERF), serão analisados os fatores de
risco apresentados abaixo, que compõe a etapa 1. </p>
        <p>Se o(a) usuário(a) apresentar apenas um fator de risco baixo/intermediário, não haverá
necessidade de calcular o escore (etapa 2), pois será considerado(a) como baixo risco
cardiovascular. </p>
        <p>Se o(a) usuário(a) apresentar ao menos um fator de alto risco, também não haverá
necessidade de calcular o escore (etapa 2), pois já será considerado(a) como alto risco
cardiovascular. </p>
        <p> O cálculo só será realizado quando o(a) usuário(a) apresentar mais de um fator de
risco
        baixo/intermediário, passando assim para a etapa 2 </p>

```

```

</div>
<div class="form-style-checkbox etapa-1">
  <h5>Etapa 1:</h5>
  <h6>Indicadores de risco baixo/intermediário</h6>
  <form action="" class="form-style" onsubmit="return false">
    <div class="style-checkbox">
      <input id="br-idade" type="checkbox" name="Idade > 65">
      <label for="br-idade">Idade > 65;</label>
    </div>
    <div class="style-checkbox">
      <input id="br-sex-masc" type="checkbox" name="Sexo masculino">
      <label for="br-sex-masc">Sexo masculino;</label>
    </div>
    <div class="style-checkbox">
      <input id="br-tabag" type="checkbox" name="Tabagismo">
      <label for="br-tabag">Tabagismo;</label>
    </div>
    <div class="style-checkbox">
      <input id="br-hipert" type="checkbox" name="Hipertensão (> 140/90 mmHg) ou
história de pré-eclâmpsia">
      <label for="br-hipert">Hipertensão (> 140/90 mmHg) ou história de pré-
eclâmpsia;</label>
    </div>
    <div class="style-checkbox">
      <input id="br-obs" type="checkbox" name="Obesidade (IMC > 30kg/m2) ou
obesidade central (cintura medida na crista ilíaca: > 88 cm em mulheres; > 102 cm em homens)">
      <label for="br-obs">Obesidade (IMC > 30kg/m2) ou obesidade central (cintura medida
na crista
ilíaca: > 88 cm em mulheres; > 102 cm em homens).</label>
    </div>
    <div class="style-checkbox">
      <input id="br-sedent" type="checkbox" name="Sedentarismo">
      <label for="br-sedent">Sedentarismo;</label>
    </div>
    <div class="style-checkbox">
      <input id="br-histo-fam" type="checkbox" name="História familiar de evento
cardiovascular prematuro em familiares de 1o grau –
pai, mãe, irmãos (homens < 55 anos e mulheres < 65 anos)">
      <label for="br-histo-fam">História familiar de evento cardiovascular prematuro em
familiares
de 1o grau –
pai, mãe, irmãos (homens < 55 anos e mulheres < 65 anos);</label>
    </div>
    <div class="style-checkbox">
      <input id="br-histo-doenca" type="checkbox" name="História de doença renal na
família (para risco de insuficiência renal)">
      <label for="br-histo-doenca">História de doença renal na família (para risco de
insuficiência renal);</label>
    </div>
    <div class="style-checkbox">
      <input id="br-manifest" type="checkbox" name="Manifestações de aterosclerose:
sopros arteriais carotídeos; diminuição ou
ausência de pulsos periféricos">
      <label for="br-manifest">Manifestações de aterosclerose: sopros arteriais carotídeos;
diminuição ou
ausência de pulsos periféricos.</label>
    </div>
  </form>
</div>

```

```

        <div class="style-checkbox">
            <input id="br-diagn-dislip" type="checkbox" name="Diagnóstico prévio de
dislipidemia">
                <label for="br-diagn-dislip">Diagnóstico prévio de dislipidemia;</label>
        </div>
        <div class="style-checkbox">
            <input id="br-diagn-polic" type="checkbox" name="Diagnóstico prévio de síndrome
de ovário policístico">
                <label for="br-diagn-polic">Diagnóstico prévio de síndrome de ovário
policístico.</label>
        </div>
        <div class="style-checkbox">
            <h5>Indicadores de alto risco</h5>
        </div>
        <div class="style-checkbox">
            <input id="ar-acid-vasc" type="checkbox" name="Acidente vascular encefálico (AVE)
prévio">
                <label for="ar-acid-vasc">Acidente vascular encefálico (AVE) prévio;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-infart-agud" type="checkbox" name="Infarto agudo do miocárdio (IAM)
prévio">
                <label for="ar-infart-agud">Infarto agudo do miocárdio (IAM) prévio;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-les-perif" type="checkbox" name="Lesão periférica – Lesão de órgão-
alvo (LOA)">
                <label for="ar-les-perif">Lesão periférica – Lesão de órgão-alvo (LOA);</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-ataq-isqu" type="checkbox" name="Ataque isquêmico transitório
(AIT)">
                <label for="ar-ataq-isqu">Ataque isquêmico transitório (AIT);</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-hiper-ventr" type="checkbox" name="Hipertrofia de ventrículo
esquerdo (HVE)">
                <label for="ar-hiper-ventr">Hipertrofia de ventrículo esquerdo (HVE);</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-nefrop" type="checkbox" name="Nefropatia">
                <label for="ar-nefrop">Nefropatia;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-retin" type="checkbox" name="Retinopatia">
                <label for="ar-retin">Retinopatia;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-aneur" type="checkbox" name="Aneurisma de aorta abdominal">
                <label for="ar-aneur">Aneurisma de aorta abdominal;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-esten" type="checkbox" name="Estenose de carótida sintomática">
                <label for="ar-esten">Estenose de carótida sintomática;</label>
        </div>
        <div class="style-checkbox">
            <input id="ar-diabet" type="checkbox" name="Diabetes mellitus">

```

```

        <label for="ar-diabet">Diabetes mellitus;</label>
    </div>
    <div class="style-checkbox">
        <input id="ar-doen-vasc" type="checkbox" name="Doença vascular periférica">
        <label for="ar-doen-vasc">Doença vascular periférica;</label>
    </div>
    <div class="style-checkbox">
        <input id="ar-insuf" type="checkbox" name="Insuficiência cardíaca congestiva de
etiologia isquêmica">
        <label for="ar-insuf">Insuficiência cardíaca congestiva de etiologia isquêmica;</label>
    </div>
    <div class="style-checkbox">
        <input id="ar-angina" type="checkbox" name="Angina no peito">
        <label for="ar-angina">Angina no peito.</label>
    </div>
    <div class="style-checkbox">
        <input id="ar-doenc-renal" type="checkbox" name="Doença renal crônica">
        <label for="ar-doenc-renal">Doença renal crônica</label>
    </div>
    <div>
        <button type="submit" id="btn-estrat-calculador">CALCULAR</button>
    </div>
</form>
<p>Classificação: <span id="retorno-calculo-estrat">Aguardando valores...</span></p>
</div>
<div class="disable-etapa-2 etapa-2">
    <div class="etapa-2-form">
        <h5>Etapa 2:</h5>
        <p>IDADE:<span id="etapa2-retorno-idade">...</span>&nbsp;SEXO:<span id="etapa2-
retorno-sexo">...</span></p>
        <form action="" class="form-style-interno" onsubmit="return false">
            <div>
                <input id="etapa2-colest-total" type="number" placeholder="COLESTEROL TOTAL">
            </div>
            <div class="style-checkbox">
                <input id="etapa2-fumante" type="checkbox" name="Fumante">
                <label for="etapa2-fumante">Fumante</label>
            </div>
            <div>
                <input id="etapa2-hdl" type="number" placeholder="HDL(MG/D)">
            </div>
            <div>
                <input id="etapa2-pa-sistol" type="number" placeholder="PA SISTÓLICA">
            </div>
            <div class="style-checkbox">
                <input id="etapa2-tratada" type="checkbox" name="Tratada">
                <label for="etapa2-tratada">Tratada</label>
            </div>
            <div>
                <button id="btn-estrat2-calculador">CALCULAR</button>
            </div>
        </form>
    </div>
    <div class="etapa-2-img">
        
    </div>
</div>

```

```

    <div class="disable-etapa-3 etapa-3">
      <h5>Etapa 3:</h5>
      <div>
        <p>Classificação: <span id="retorno-calculo-estrat3">Aguardando valores...</span></p>
      </div>
    </div>
  </div>
  <div>
    <a class="botao-padrao">
      <button id="btn-finaliza-ava-clinic">FINALIZAR AVALIAÇÃO</button>
    </a>
  </div>
  <p id="retorno-finaliza-ava-clinic">...</p>
</div>
</div>
<script src="../js/retonoDados/retornoDadosAvaClinic.js"></script>
<script type="module" src="../js/insertDadosStorageAvaClinic.js"></script>
<script src="../js/Utils/calculos.js"></script>
<script type="module" src="../js/activeButton/activeButtonAva.js"></script>
<script src="../js/finalizaAvaClinic.js"></script>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
<script type="text/javascript" src="../js/jquery-3.6.4.min.js"></script>
<script type="text/javascript" src="../js/jquery.mask.js"></script>
<script type="text/javascript">
  $(document).ready(function () {
    $('#imc-peso').mask('#0.00', { reverse: true })
    $('#imc-altura').mask('#0.00', { reverse: true })
  });
</script>
</body>
</html>

```

pages/pageAvaClinicExemplo.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/displayNone.css">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.16/jquery.mask.js"
  integrity="sha256-yE5LLp5HSQ/z+hJeCqkz9hdjNkk1jaiGG0tDCraumnA="
  crossorigin="anonymous"></script>
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">

```

```

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <div>
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/adm_area.html">
                        
                        <p class="text-menu"> MENU INICIAL</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page" href="/pageHistoricoEnf.html">
                        
                        <p class="text-menu"> GUIA PE</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" aria-current="page" href="/menu2.html">
                        
                        <p class="text-menu"> REALIZACAO PE</p>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link btn-logout" href="/cadastro.html">
                        
                        <p class="text-menu"> SAIR</p>
                    </a>
                </li>
            </ul>
        </div>
    </div>
</div>
</div>
</div>
</div>
<div class="fundo-botao">
    <a class="botao-padrao">
        <button id="btn-rastreamento"> RASTREAMENTO DA HIPERTENSÃO ARTERIAL<span
class="icon-expand"></span></button>
    </a>
    <div class="text-none-rastreamento text-none-style">
        <div>
            <h4> Rastreamento da hipertensão arterial</h4>
            <div class="text-style-pure2">
                <p> Para o rastreamento recomenda-se a verificação na Unidade pelo menos duas vezes
na
                semana, por duas semanas e se possível, conciliar com a Monitorização Residencial da
                Pressão Arterial (MRPA), sendo duas vezes ao dia durante sete dias ou três vezes ao
dia
                durante cinco dias (PROTOCOLO DE ENFERMAGEM: HIPERTENSÃO E
                DIABETES, 2021).</p>
            </div>
        </div>
    </div>

```



```

<div>
  <h4>Valores aferidos</h4>
  <form class="form-style" onsubmit="return false">
    <div>
      <input id="pas" type="number" placeholder="PAS: MMHG" readonly>
    </div>
    <div>
      <input id="pad" type="number" placeholder="PAD: MMHG" readonly>
    </div>
    <div>
      <button id="btn-calcular-pas-pad">CALCULAR</button readonly>
    </div>
  </form>
  <p>Classificação: <span id="retorno-calculo-pas-pad">SOMENTE EXEMPLO</span></p>
</div>
</div>
<a class="botao-padrao">
  <button id="btn-antropom">DADOS ANTROPOMÉTRICOS<span class="icon-expand"></span></button>
</a>
<div class="text-none-antropom text-none-style">
  <div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
      <button id="btn-imc">ÍNDICE DE MASSA CORPORAL (IMC)<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-imc text-none-style-interno">
      <h4>ÍNDICE DE MASSA CORPORAL (IMC)</h4>
      <div>
        <div>
          <p>IDADE:<span id="imc-retorno-idade">...</span>&nbsp;SEXO:<span id="imc-
retorno-sexo">...</span></p>
        </div>
        <form action="" class="form-style-interno" onsubmit="return false">
          <div>
            <input id="imc-peso" type="text" placeholder="PESO(KG): 80" readonly>
          </div>
          <div>
            <input id="imc-altura" type="text" placeholder="ALTURA(M): 1.70" readonly>
          </div>
          <div>
            <button id="btn-imc-calcular">CALCULAR</button readonly>
          </div>
        </form>
        <p>Classificação: <span id="retorno-calculo-imc">SOMENTE EXEMPLO</span></p>
      </div>
    </div>
  </div>
</div>
<div class="flex-ajuste">
  <a class="botao-padrao botao-padrao-interno">
    <button id="btn-circ-abdom">CIRCUNFERÊNCIA ABDOMINAL<span class="icon-
expand"></span></button>
  </a>
  <div class="text-none-circ-abdom text-none-style-interno">
    <h4>CIRCUNFERÊNCIA ABDOMINAL</h4>
    <div>
      <div>

```

```

        <p>SEXO:<span id="circ-abdom-retorno-sexo">...</span></p>
    </div>
    <form action="" class="form-style-interno" onsubmit="return false">
        <div>
            <input id="circ-abdom-cm" type="number" placeholder="(CM)" readonly>
        </div>
        <div>
            <button id="btn-circ-abdom-calcul" >CALCULAR</button>
        </div>
    </form>
    <p>Classificação: <span id="retorno-calculo-circ-abdom">SOMENTE
EXEMPLO</span></p>
    </div>
</div>
<div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
        <button id="btn-cint-quadr">RELAÇÃO CINTURA-QUADRIL<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-cint-quadr text-none-style-interno">
        <h4>RELAÇÃO CINTURA-QUADRIL</h4>
        <div>
            <div>
                <p>SEXO:<span id="cint-quadr-retorno-sexo">...</span></p>
            </div>
            <form action="" class="form-style-interno" onsubmit="return false">
                <div>
                    <input id="cint-quadr-abdom-cm" type="number" placeholder="(CM)
CIRCUNFÊRENCIA ABDOMINAL" readonly>
                </div>
                <div>
                    <input id="cint-quadr-quadr-cm" type="number" placeholder="(CM)
CIRCUNFERÊNCIA DO QUADRIL" readonly>
                </div>
                <div>
                    <button id="btn-cint-quadr-calcul" >CALCULAR</button>
                </div>
            </form>
            <p>Classificação: <span id="retorno-calculo-cint-quadr">SOMENTE
EXEMPLO</span></p>
        </div>
    </div>
</div>
<div class="flex-ajuste">
    <a class="botao-padrao botao-padrao-interno">
        <button id="btn-torn-braq">ÍNDICE TORNOZELO-BRAQUIAL<span class="icon-
expand"></span></button>
    </a>
    <div class="text-none-torn-braq text-none-style-interno">
        <h4>ÍNDICE TORNOZELO-BRAQUIAL</h4>
        <div>
            <form action="" class="form-style-interno" onsubmit="return false">
                <div>
                    <input id="torn-braq-pas-torn" type="number" placeholder="VALOR DA MAIOR
PAS DO TORNOZELO" readonly>
                </div>

```

```

        <div>
            <input id="torn-braq-pas-bra" type="number" placeholder="VALOR DA PAS DO
BRAÇO" readonly>
        </div>
        <div>
            <button id="btn-torn-braq-calcul" >CALCULAR</button>
        </div>
    </form>
    <p>Classificação: <span id="retorno-calculo-torn-braq">SOMENTE
EXEMPLO</span></p>
</div>
</div>
</div>
<div class="botao-padrao">
    <button id="btn-estrat-risco">ESTRATIFICAÇÃO DO RISCO CARDIOVASCULAR<span
class="icon-expand"></span></button>
</a>
<div class="text-none-estrat-risco text-none-style">
    <h4>ESTRATIFICAÇÃO DO RISCO CARDIOVASCULAR</h4>
    <div class="text-style-pure2">
        <p> Para o cálculo da Escala de Risco de Framingham (ERF), serão analisados os fatores de
risco apresentados abaixo, que compõe a etapa 1. </p>
        <p>Se o(a) usuário(a) apresentar apenas um fator de risco baixo/intermediário, não haverá
necessidade de calcular o escore (etapa 2), pois será considerado(a) como baixo risco
cardiovascular. </p>
        <p>Se o(a) usuário(a) apresentar ao menos um fator de alto risco, também não haverá
necessidade de calcular o escore (etapa 2), pois já será considerado(a) como alto risco
cardiovascular. </p>
        <p> O cálculo só será realizado quando o(a) usuário(a) apresentar mais de um fator de
risco
baixo/intermediário, passando assim para a etapa 2 </p>
    </div>
<div class="form-style-checkbox etapa-1">
    <h5>Etapa 1:</h5>
    <h6>Indicadores de risco baixo/intermediário</h6>
    <form action="" class="form-style" onsubmit="return false">
        <div class="style-checkbox" readonly>
            <input id="br-idade" type="#" name="Idade > 65" readonly>
            <label for="br-idade">Idade > 65;</label>
        </div>
        <div class="style-checkbox" readonly>
            <input id="br-sex-masc" type="#" name="Sexo masculino" readonly>
            <label for="br-sex-masc">Sexo masculino;</label>
        </div>
        <div class="style-checkbox" readonly>
            <input id="br-tabag" type="#" name="Tabagismo" readonly>
            <label for="br-tabag">Tabagismo;</label>
        </div>
        <div class="style-checkbox" readonly>
            <input id="br-hipert" type="#" name="Hipertensão (> 140/90 mmHg) ou história de
pré-eclâmpsia" readonly>
            <label for="br-hipert">Hipertensão (> 140/90 mmHg) ou história de pré-
eclâmpsia;</label>
        </div>
    </form>
</div>

```

<label for="br-obs">Obesidade (IMC > 30kg/m2) ou obesidade central (cintura medida na crista ilíaca: > 88 cm em mulheres; > 102 cm em homens).</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-sedent" type="#" name="Sedentarismo" readonly>
 <label for="br-sedent">Sedentarismo;</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-histo-fam" type="#" name="História familiar de evento cardiovascular prematuro em familiares de 1o grau – pai, mãe, irmãos (homens < 55 anos e mulheres < 65 anos)" readonly>
 <label for="br-histo-fam">História familiar de evento cardiovascular prematuro em familiares de 1o grau – pai, mãe, irmãos (homens < 55 anos e mulheres < 65 anos);</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-histo-doenca" type="#" name="História de doença renal na família (para risco de insuficiência renal)">
 <label for="br-histo-doenca" readonly>História de doença renal na família (para risco de insuficiência renal);</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-manifest" type="#" name="Manifestações de aterosclerose: sopros arteriais carotídeos; diminuição ou ausência de pulsos periféricos" readonly>
 <label for="br-manifest">Manifestações de aterosclerose: sopros arteriais carotídeos; diminuição ou ausência de pulsos periféricos.</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-diagn-dislip" type="#" name="Diagnóstico prévio de dislipidemia" readonly>
 <label for="br-diagn-dislip">Diagnóstico prévio de dislipidemia;</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="br-diagn-polic" type="#" name="Diagnóstico prévio de síndrome de ovário policístico" readonly>
 <label for="br-diagn-polic">Diagnóstico prévio de síndrome de ovário policístico.</label readonly>
 </div>
 <div class="style-checkbox">
 <h5>Indicadores de alto risco</h5>
 </div>
 <div class="style-checkbox" readonly>
 <input id="ar-acid-vasc" type="#" name="Acidente vascular encefálico (AVE) prévio" readonly>
 <label for="ar-acid-vasc">Acidente vascular encefálico (AVE) prévio;</label readonly>
 </div>
 <div class="style-checkbox" readonly>
 <input id="ar-infert-agud" type="#" name="Infarto agudo do miocárdio (IAM) prévio" readonly>
 <label for="ar-infert-agud">Infarto agudo do miocárdio (IAM) prévio;</label readonly>
 </div>

```

</div>
<div class="style-checkbox" readonly>
  <input id="ar-les-perif" type="#" name="Lesão periférica – Lesão de órgão-alvo (LOA)"
readonly>
  <label for="ar-les-perif">Lesão periférica – Lesão de órgão-alvo (LOA);</label
readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-ataq-isqu" type="#" name="Ataque isquêmico transitório (AIT)"
readonly>
  <label for="ar-ataq-isqu">Ataque isquêmico transitório (AIT);</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-hiper-ventr" type="#" name="Hipertrofia de ventrículo esquerdo (HVE)"
readonly>
  <label for="ar-hiper-ventr">Hipertrofia de ventrículo esquerdo (HVE);</label
readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-nefrop" type="#" name="Nefropatia" readonly>
  <label for="ar-nefrop">Nefropatia;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-retin" type="#" name="Retinopatia" readonly>
  <label for="ar-retin">Retinopatia;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-aneur" type="#" name="Aneurisma de aorta abdominal" readonly>
  <label for="ar-aneur">Aneurisma de aorta abdominal;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-esten" type="#" name="Estenose de carótida sintomática" readonly>
  <label for="ar-esten">Estenose de carótida sintomática;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-diabet" type="#" name="Diabetes mellitus" readonly>
  <label for="ar-diabet">Diabetes mellitus;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-doen-vasc" type="#" name="Doença vascular periférica" readonly>
  <label for="ar-doen-vasc">Doença vascular periférica;</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-insuf" type="#" name="Insuficiência cardíaca congestiva de etiologia
isquêmica" readonly>
  <label for="ar-insuf">Insuficiência cardíaca congestiva de etiologia isquêmica;</label
readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-angina" type="#" name="Angina no peito" readonly>
  <label for="ar-angina">Angina no peito.</label readonly>
</div>
<div class="style-checkbox" readonly>
  <input id="ar-doenc-renal" type="#" name="Doença renal crônica" readonly>
  <label for="ar-doenc-renal">Doença renal crônica</label readonly>
</div>
</div>

```

```

        <button type="submit" id="#">CALCULAR</button readonly>
    </div>
</form>
<p>Classificação: <span id="#">SOMENTE EXEMPLO.</span></p readonly>
</div>
<div class="disable-etapa-2 etapa-2">
    <div class="etapa-2-form">
        <h5>Etapa 2:</h5>
        <p>IDADE:<span id="etapa2-retorno-idade">...</span>&emsp;SEXO:<span id="etapa2-
retorno-sexo">...</span></p>
        <form action="" class="form-style-interno" onsubmit="return false">
            <div>
                <input id="etapa2-colest-total" type="number" placeholder="COLESTEROL TOTAL">
            </div>
            <div class="style-checkbox">
                <input id="etapa2-fumante" type="#" name="Fumante">
                <label for="etapa2-fumante">Fumante</label readonly>
            </div>
            <div>
                <input id="etapa2-hdl" type="number" placeholder="HDL(MG/D)" readonly>
            </div>
            <div>
                <input id="etapa2-pa-sistol" type="number" placeholder="PA SISTÓLICA" readonly>
            </div>
            <div class="style-checkbox">
                <input id="etapa2-tratada" type="#" name="Tratada" readonly>
                <label for="etapa2-tratada">Tratada</label>
            </div>
            <div>
                <button id="btn-estrat2-calcular">CALCULAR</button readonly>
            </div>
        </form>
    </div>
    <div class="etapa-2-img">
        
    </div>
</div>
<div class="disable-etapa-3 etapa-3">
    <h5>Etapa 3:</h5>
    <div>
        <p>Classificação: <span id="#">SOMENTE EXEMPLO</span></p readonly>
    </div>
</div>
</div>
<div>
    <a class="botao-padrao">
        <button id="#">FINALIZAR AVALIAÇÃO</button readonly>
    </a>
</div>
<p id="retorno-finaliza-ava-clinic">...</p readonly>
</div>
</div>
<script src="../../js/retonoDados/retornoDadosAvaClinic.js" readonly></script>
<script type="module" src="../../js/insertDadosStorageAvaClinic.js" readonly></script>
<script src="../../js/Utils/calculos.js" readonly></script>
<script type="module" src="../../js/activeButton/activeButtonAva.js" readonly></script>
<script src="../../js/finalizaAvaClinic.js" readonly></script>

```

```

<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js" > readonly</script>
<script type="text/javascript" src="../../js/jquery-3.6.4.min.js"></script>
<script type="text/javascript" src="../../js/jquery.mask.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $('#imc-peso').mask('#0.00', { reverse: true })
        $('#imc-altura').mask('#0.00', { reverse: true })
    });
</script>
</body>

</html>

```

pages/pageDiagnosEnf.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">
            <nav class="navbar navbar-expand-lg">
                <div class="container-fluid">
                    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="collapse navbar-collapse" id="navbarNav">
                        <div>
                            <ul class="navbar-nav">
                                <li class="nav-item">
                                    <a class="nav-link" href="../../adm_area.html">
                                        
                                        <p class="text-menu">MENU INICIAL</p>
                                    </a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link" aria-current="page" href="../../menu1.html">
                                        
                                        <p class="text-menu">GUIA PE</p>
                                    </a>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link active" aria-current="page" href="../../menu2.html">
                                        

```

```

        <p class="text-menu">REALIZACAO PE</p>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link btn-logout" href="/cadastro.html">
        
        <p class="text-menu">SAIR</p>
    </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="fundo-botao">
    <div class="text-style-pure2">
        <p>Os Diagnósticos/Resultados de Enfermagem apresentados abaixo consideram a
        Classificação Internacional para a Prática de Enfermagem (CIPE®), especificamente o
        Subconjunto Terminológico da CIPE® para pessoas com hipertensão na Atenção
        Primária a Saúde e estão organizados conforme a Teoria das Necessidades Humanas
        Básicas de Wanda de Aguiar Horta.
    </p>
</div>
<a href="/pagesNecessidades/pageNecessBiolog.html" class="botao-padrao">
    <button>NECESSIDADES HUMANAS BÁSICAS (NHB) - PSICOBIOLOGICAS</button>

</a>
<a href="/pagesNecessidades/pageNecessSocia.html" class="botao-padrao">
    <button>NECESSIDADES HUMANAS BÁSICAS (NHB) - PSICOSOCIAIS</button>
</a>
<a href="/pagesNecessidades/pageNecessEspir.html" class="botao-padrao">
    <button>NECESSIDADES HUMANAS BÁSICAS (NHB) - PSICOESPIRITUAIS</button>

</a>
</div>
</div>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageDireitos.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../css/global.css">
    <title>HAsupport</title>
</head>

<body>

```



```

<div class="fundo1 pageDireitos">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="/cadastro.html">
                  
                  <p class="text-menu">VOLTAR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </div>
    <div class="fundo-botao">
      <html>

      <head>
        <style>
          h1 {
            font-weight: bold;
            text-align: right;
            width: 30%;
          }

          h4 {
            font-weight: bold;
            text-align: left;
            width: 60%;
          }

          h5 {
            text-align: justify;
            width: 60%;
          }
        </style>
      </head>

      <h1>Referências:</h1>
      <br>
      <h5> BARROS, Alba Lucia Bottura Leite Anamnese e Exame Físico: avaliação diagnóstica
de enfermagem no adulto. 4. ed. Porto Alegre: Artmed, 2022.
      </h5>
      <br>
      <h5> BARROSO, Weimar Kunz Sebba et al. Diretrizes Brasileiras de Hipertensão Arterial –
2020. Arquivos Brasileiros de Cardiologia, v. 116, n.3, p.516-658, 2021. doi:
https://doi.org/10.36660/abc.20201238

```

</h5>

<br>

<h5> CONSELHO FEDERAL DE ENFERMAGEM. Resolução nº 358, de 15 de outubro de 2009. Dispõe sobre a Sistematização da Assistência de Enfermagem - SAE – em ambientes públicos e privados em que ocorre o cuidado profissional de Enfermagem e da outras providencias. Brasília, DF, 2009.

</h5>

<br>

<h5> GARCIA, Telma Ribeiro. (org.). Classificação Internacional para a Prática de Enfermagem – CIPE®. Porto Alegre: Artmed, 2020.

</h5>

<br>

<h5> NÓBREGA, Renata Valéria. Proposta de subconjunto terminológico da Classificação Internacional para a Prática de Enfermagem (CIPE®) para hipertensos na Atenção Básica. 2012. Dissertação (Mestrado em Enfermagem) - Universidade Federal da Paraíba. João Pessoa, PB, 2012.

</h5>

<br>

<h5> PROTOCOLO DE ENFERMAGEM: HIPERTENSÃO E DIABETES. Organização: Secretaria Municipal de Saúde de Santa Maria. Adaptado do Protocolo do COREN-RS, 2020. Santa Maria, 2021.

</h5>

<br>

<h5> PROTOCOLO DE ENFERMAGEM: HIPERTENSÃO E DIABETES. Organização: Secretaria Municipal de Saúde de Santa Maria. Adaptado do Protocolo do COREN-RS, 2020. Santa Maria, 2021.

</h5>

<br>

<h5> PROTOCOLO DE ENFERMAGEM: HIPERTENSÃO E DIABETES. Organização: Secretaria Municipal de Saúde de Santa Maria. Adaptado do Protocolo do COREN-RS, 2020. Santa Maria, 2021.

</h5>

<br>

<br>

<h1>Elaboração:</h1>

<br>

<h4>Equipe de Enfermagem:</h4>

<h5>Enfa. Dda. Gabriela Oliveira</h5>

<h5>Profa. Dra. Maria Denise Schimith</h5>

<h5>Profa. Dra. Cândida Caniçali Primo</h5>

<br>

<h4>Equipe de Desenvolvimento:</h4>

<h5>Eduardo Santos de Paula</h5>

<h5>Fábio Corrêa Costa Junior</h5>

<h5>Lucas Ziquinatti Piccini Nunes</h5>

<h5>Winglisson Maretoli Gonçalves</h5>

<br>

<h4>Equipe de Supervisão e Orientação do Desenvolvimento:</h4>

<h5>Prof. Dr. Daniel Welfer</h5>

<h5>Prof. Dr. Joaquim V. C. Assunção</h5>

<h5>Profa. Dra. Andrea Schwertner Charão</h5>

<br>

<h4>Apoio:</h4>

<h5>Integrantes do Grupo de Pesquisa: Cuidado, Saúde e Enfermagem – Núcleo de Ensino, Pesquisa e Extensão Práticas de Cuidado nos Cenários da Saúde (PraCCeS).

```

    </h5>
    <h5>Programa de Pós-Graduação em Enfermagem (PPGENF) da Universidade Federal de
    Santa Maria (UFSM).</h5>
    <h5>Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)</h5>
    <h5>Secretaria de Município da Saúde — Santa Maria</h5>
  </div>
</div>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageExamesLab.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>

```

```

    </li>
    <li class="nav-item">
      <a class="nav-link btn-logout" href="/cadastro.html">
        
        <p class="text-menu">SAIR

        </p>
      </a>
    </li>
  </ul>
</div>
</div>
</div>
</nav>
</div>
<div class="fundo-botao">
  <form class="form-style form-style-hemograma" onsubmit="return false">
    <h5>HEMOGRAMA</h5>
    <div>
      <label for="hemog-hemac">Hemácias</label>
      <input id="hemog-hemac" type="number" placeholder="4.58 milhões/mm3">
    </div>
    <div>
      <label for="hemog-hemogl">Hemoglobina</label>
      <input id="hemog-hemogl" type="number" placeholder="14.2 g/dL">
    </div>
    <div>
      <label for="hemog-hemat">Hematócrito</label>
      <input id="hemog-hemat" type="number" placeholder="41,8%">
    </div>
    <div>
      <label for="hemog-leuc">Leucócitos</label>
      <input id="hemog-leuc" type="number" placeholder="8.300/mm3 (Segmentados 51,0%;
Eosinófilos 3,0%; Basófilos 0,0%)">
    </div>
    <div>
      <label for="hemog-linf">Linfócitos</label>
      <input id="hemog-linf" type="number" placeholder="39,0%; Monócitos 7,0%">
    </div>
    <div>
      <label for="hemog-plaq">Plaquetas</label>
      <input id="hemog-plaq" type="number" placeholder="327.000/mm3">
    </div>
    <div>
      <label for="hemog-glic">Glicemia de jejum</label>
      <input id="hemog-glic" type="number" placeholder="89 md/dL">
    </div>
    <div>
      <label for="hemog-coles">Colesterol total</label>
      <input id="hemog-coles" type="number" placeholder="206 mg/dL">
    </div>
    <div>
      <label for="hemog-hdl">HDL</label>
      <input id="hemog-hdl" type="number" placeholder="84 mg/dL">
    </div>
    <div>
      <label for="hemog-trig">Triglicerídios</label>

```

```

        <input id="hemog-trig" type="number" placeholder="75 mg/dL">
    </div>
    <div>
        <label for="hemog-hemog-glic">Hemoglobina Glicada</label>
        <input id="hemog-hemog-glic" type="number" placeholder="5,6%">
    </div>
    <div>
        <label for="hemog-creat">Creatina</label>
        <input id="hemog-creat" type="number" placeholder="0,80 mg/dL">
    </div>
    <div>
        <label for="hemog-potass">Potássio Plasmático</label>
        <input id="hemog-potass" type="number" placeholder="4,2 mmol/L">
    </div>
    <div>
        <label for="hemog-equ">EQU</label>
        <input id="hemog-equ" type="text" placeholder="Exemplo: Proteínas - positivo">
    </div>
    <div>
        <label for="hemog-microa">Microalbumina na urina</label>
        <input id="hemog-microa" type="number" placeholder="18 mg/L">
    </div>
    <div>
        <label for="hemog-album">Albumina</label>
        <input id="hemog-album" type="number" placeholder="4.8 g/dL">
    </div>
    <div>
        <label for="hemog-acid">Ácido úrico</label>
        <input id="hemog-acid" type="number" placeholder="2.8 mg/dL">
    </div>
    <div>
        <label for="hemog-ureia">Ureia</label>
        <input id="hemog-ureia" type="number" placeholder="34 mg/dL">
    </div>
    <div>
        <label for="hemog-tgo">TGO</label>
        <input id="hemog-tgo" type="number" placeholder="20 U/L">
    </div>
    <div>
        <label for="hemog-tgp">TGP</label>
        <input id="hemog-tgp" type="number" placeholder="40 U/L">
    </div>
    <div>
        <button id="btn-hemog-calcular">SALVAR HEMOGRAMA</button>
    </div>
    <div>
        <p id="retorno-salvo-hemograma">...</p>
    </div>
</form>
</div>
</div>
<script type="module" src="../../js/insertDadosStorageHemog.js"></script>
<script src="../../js/retonoDados/retornoDadosHemog.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

</html>

pages/pageExamesLabExemplo.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

<link rel="stylesheet" href="../css/global.css">

<title>HAsupport</title>

</head>

<body>

<div class="fundo1">

<div class="menu">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="/adm\_area.html">



<p class="text-menu">MENU INICIAL</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="/pageHistoricoEnf.html">



<p class="text-menu">GUIA PE</p>

</a>

</li>

</li>

<li class="nav-item">

<a class="nav-link " aria-current="page" href="/menu2.html">



<p class="text-menu">REALIZACAO PE</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link btn-logout" href="/cadastro.html">



<p class="text-menu">SAIR

</p>

</a>

</li>

</ul>

```
</div>
</div>
</div>
</nav>
</div>
<div class="fundo-botao">
<form class="form-style form-style-hemograma" onsubmit="return false">
<h5>HEMOGRAMA</h5>
<div>
<label for="hemog-hemac">Hemácias</label>
<input id="hemog-hemac" type="number" placeholder="4.58 milhões/mm3" readonly>
</div>
<div>
<label for="hemog-hemogl">Hemoglobina</label>
<input id="hemog-hemogl" type="number" placeholder="14.2 g/dL" readonly>
</div>
<div>
<label for="hemog-hemat">Hematócrito</label>
<input id="hemog-hemat" type="number" placeholder="41,8%" readonly>
</div>
<div>
<label for="hemog-leuc">Leucócitos</label>
<input id="hemog-leuc" type="number" placeholder="8.300/mm3 (Segmentados 51,0%;
Eosinófilos 3,0%; Basófilos 0,0%)" readonly>
</div>
<div>
<label for="hemog-linf">Linfócitos</label>
<input id="hemog-linf" type="number" placeholder="39,0%; Monócitos 7,0%" readonly>
</div>
<div>
<label for="hemog-plaq">Plaquetas</label>
<input id="hemog-plaq" type="number" placeholder="327.000/mm3" readonly>
</div>
<div>
<label for="hemog-glic">Glicemia de jejum</label>
<input id="hemog-glic" type="number" placeholder="89 md/dL" readonly>
</div>
<div>
<label for="hemog-coles">Colesterol total</label>
<input id="hemog-coles" type="number" placeholder="206 mg/dL" readonly>
</div>
<div>
<label for="hemog-hdl">HDL</label>
<input id="hemog-hdl" type="number" placeholder="84 mg/dL" readonly>
</div>
<div>
<label for="hemog-trig">Triglicerídios</label>
<input id="hemog-trig" type="number" placeholder="75 mg/dL" readonly>
</div>
<div>
<label for="hemog-hemog-glic">Hemoglobina Glicada</label>
<input id="hemog-hemog-glic" type="number" placeholder="5,6%" readonly>
</div>
<div>
<label for="hemog-creat">Creatina</label>
<input id="hemog-creat" type="number" placeholder="0,80 mg/dL" readonly>
</div>
```

```

<div>
  <label for="hemog-potass">Potássio Plasmático</label>
  <input id="hemog-potass" type="number" placeholder="4,2 mmol/L" readonly>
</div>
<div>
  <label for="hemog-equ">EQU</label>
  <input id="hemog-equ" type="text" placeholder="Exemplo: Proteínas - positivo" readonly>
</div>
<div>
  <label for="hemog-microa">Microalbumina na urina</label>
  <input id="hemog-microa" type="number" placeholder="18 mg/L" readonly>
</div>
<div>
  <label for="hemog-album">Albumina</label>
  <input id="hemog-album" type="number" placeholder="4.8 g/dL" readonly>
</div>
<div>
  <label for="hemog-acid">Ácido úrico</label>
  <input id="hemog-acid" type="number" placeholder="2.8 mg/dL" readonly>
</div>
<div>
  <label for="hemog-ureia">Ureia</label>
  <input id="hemog-ureia" type="number" placeholder="34 mg/dL" readonly>
</div>
<div>
  <label for="hemog-tgo">TGO</label>
  <input id="hemog-tgo" type="number" placeholder="20 U/L" readonly>
</div>
<div>
  <label for="hemog-tgp">TGP</label>
  <input id="hemog-tgp" type="number" placeholder="40 U/L" readonly>
</div>
<div>
  <button id="btn-hemog-calcular">SALVAR HEMOGRAMA </button readonly>
</div>
<div>
  <p id="retorno-salvo-hemograma">...</p>
</div>
</form>
</div>
</div>
<script type="module" src="../js/insertDadosStorageHemog.js"></script>
<script src="../js/retonoDados/retornoDadosHemog.js"></script>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageExamesLabExemplo.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">

```



```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
<link rel="stylesheet" href="../css/global.css">
<title>HAsupport</title>
</head>

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="/adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/pageHistoricoEnf.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link " aria-current="page" href="/menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="/cadastro.html">
                  
                  <p class="text-menu">SAIR

                </p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="fundo-botao">
    <form class="form-style form-style-hemograma" onsubmit="return false">
      <h5>HEMOGRAMA</h5>
      <div>
        <label for="hemog-hemac">Hemácias</label>

```

```
<input id="hemog-hemac" type="number" placeholder="4.58 milhões/mm3" readonly>
</div>
<div>
  <label for="hemog-hemogl">Hemoglobina</label>
  <input id="hemog-hemogl" type="number" placeholder="14.2 g/dL" readonly>
</div>
<div>
  <label for="hemog-hemat">Hematócrito</label>
  <input id="hemog-hemat" type="number" placeholder="41,8%" readonly>
</div>
<div>
  <label for="hemog-leuc">Leucócitos</label>
  <input id="hemog-leuc" type="number" placeholder="8.300/mm3 (Segmentados 51,0%;
Eosinófilos 3,0%; Basófilos 0,0%)" readonly>
</div>
<div>
  <label for="hemog-linf">Linfócitos</label>
  <input id="hemog-linf" type="number" placeholder="39,0%; Monócitos 7,0%" readonly>
</div>
<div>
  <label for="hemog-plaq">Plaquetas</label>
  <input id="hemog-plaq" type="number" placeholder="327.000/mm3" readonly>
</div>
<div>
  <label for="hemog-glic">Glicemia de jejum</label>
  <input id="hemog-glic" type="number" placeholder="89 md/dL" readonly>
</div>
<div>
  <label for="hemog-coles">Colesterol total</label>
  <input id="hemog-coles" type="number" placeholder="206 mg/dL" readonly>
</div>
<div>
  <label for="hemog-hdl">HDL</label>
  <input id="hemog-hdl" type="number" placeholder="84 mg/dL" readonly>
</div>
<div>
  <label for="hemog-trig">Triglicerídios</label>
  <input id="hemog-trig" type="number" placeholder="75 mg/dL" readonly>
</div>
<div>
  <label for="hemog-hemog-glic">Hemoglobina Glicada</label>
  <input id="hemog-hemog-glic" type="number" placeholder="5,6%" readonly>
</div>
<div>
  <label for="hemog-creat">Creatina</label>
  <input id="hemog-creat" type="number" placeholder="0,80 mg/dL" readonly>
</div>
<div>
  <label for="hemog-potass">Potássio Plasmático</label>
  <input id="hemog-potass" type="number" placeholder="4,2 mmol/L" readonly>
</div>
<div>
  <label for="hemog-equ">EQU</label>
  <input id="hemog-equ" type="text" placeholder="Exemplo: Proteínas - positivo" readonly>
</div>
<div>
  <label for="hemog-microa">Microalbumina na urina</label>
```

```

        <input id="hemog-microa" type="number" placeholder="18 mg/L" readonly>
    </div>
    <div>
        <label for="hemog-album">Albumina</label>
        <input id="hemog-album" type="number" placeholder="4.8 g/dL" readonly>
    </div>
    <div>
        <label for="hemog-acid">Ácido úrico</label>
        <input id="hemog-acid" type="number" placeholder="2.8 mg/dL" readonly>
    </div>
    <div>
        <label for="hemog-ureia">Ureia</label>
        <input id="hemog-ureia" type="number" placeholder="34 mg/dL" readonly>
    </div>
    <div>
        <label for="hemog-tgo">TGO</label>
        <input id="hemog-tgo" type="number" placeholder="20 U/L" readonly>
    </div>
    <div>
        <label for="hemog-tgp">TGP</label>
        <input id="hemog-tgp" type="number" placeholder="40 U/L" readonly>
    </div>
    <div>
        <button id="btn-hemog-calcular">SALVAR HEMOGRAMA </button readonly>
    </div>
    <div>
        <p id="retorno-salvo-hemograma">...</p>
    </div>
</form>
</div>
</div>
<script type="module" src="../../js/insertDadosStorageHemog.js"></script>
<script src="../../js/retonoDados/retornoDadosHemog.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageFinalizar.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <title>HAsupport</title>
</head>

<body>
    <div class="fundo1">
        <div class="menu">

```

```

<nav class="navbar navbar-expand-lg">
  <div class="container-fluid">
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <div>
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="/adm_area.html">
              
              <p class="text-menu">MENU INICIAL</p>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" aria-current="page" href="/menu1.html">
              
              <p class="text-menu">GUIA PE</p>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" aria-current="page" href="/menu2.html">
              
              <p class="text-menu">REALIZACAO PE</p>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link btn-logout" href="/cadastro.html">
              
              <p class="text-menu">SAIR</p>
            </a>
          </li>
        </ul>
      </div>
    </div>
  </div>
</nav>
</div>
<div class="funcao-botao">
  <div>
    <a class="botao-padrao">
      <button id="btn-finaliza-consulta">FINALIZAR CONSULTA</button>
    </a>
  </div>
  <p id="retorno-finaliza-cosulta">...</p>
</div>
</div>
<script type="module" src="../backend/connection/atendimento_db.js"></script>
<script src="../js/finalizaConsulta.js"></script>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageGuiaEntre.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../css/global.css">
  <link rel="stylesheet" href="../css/displayNone.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="/adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="/pageHistoricoEnf.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="/cadastro.html">
                    
                    <p class="text-menu">SAIR</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>
</body>
</html>
```

```

    </div>
  </nav>
</div>
<div class="fundo-botao">
  <a class="botao-padrao">
    <button id="btn-obj-entrevista">OBJETIVOS PARA ENTREVISTA<span class="icon-expand"></span></button>
  </a>
  <div class="text-none-obj-entrevista text-none-style">
    <div class="ajuste-text">
      <p>1) Saber como o(a) usuário(a) está – condições físicas e emocionais.</p>
      <p>2) Situar como o(a) usuário(a) é – características gerais e seus hábitos.</p>
      <p>3) Compreender como o(a) usuário(a) percebe o processo saúde-doença, bem como as
crenças e valores.</p>
      <p>4) Identificar as demandas de cuidado percebidas pelo(a) usuário(a) e enfermeira(o).</p>
      <p>5) Identificar sinais e sintomas de alterações fisiológicas, emocionais, mentais, espirituais
e sociais.</p>
    </div>
  </div>
  <a href="#" class="botao-padrao">
    <button id="btn-quest-import">QUESTIONAMENTOS IMPORTANTES<span class="icon-
expand"></span></button>
  </a>
  <div class="text-none-quest-import text-none-style">
    <div class="ajuste-text">
      <p>Motivo da procura do serviço e/ou queixa principal.</p>
      <p>Presença de doenças e tratamentos anteriores, bem como alergias.</p>
      <p>Antecedentes familiares (familiares em linhagem direta (pais, avós, tios e irmãos) óbito
por motivos de doenças cardiovasculares ou possuem diagnóstico de alguma doença
crônica).</p>
      <p>Uso de medicamentos (medicamentos utilizados em casa tanto os prescritos pelo(a)
profissional médico(a) como os não prescritos, investigar as doses, horários, uso regular
ou não).</p>
      <p>Existência de fatores de risco (diagnóstico de outra doença crônica, como diabetes
mellitus, presença de outros fatores de risco para doenças cardiovasculares, por exemplo,
tabagismo, sedentarismo, sobrepeso ou obesidade, dislipidemia. Para as mulheres
considerar o uso de anticoncepcional ou reposição hormonal).</p>
      <p>Hábitos e costumes (dieta, exercício físico, sedentarismo, estressores diários, forma de
relaxamento e lazer).</p>
      <p>Nesse momento, torna-se importante atentar e investigar as manifestações clínicas
mais comuns das doenças cardiovasculares e fatores de risco como a hipertensão arterial,
tais como:</p>
      <p class="ajuste-text-destaque">Falta de ar (dispneia); Fadiga; Dor no peito (precordialgia);
Desconforto no peito;
      Palpitações; Desmaio; Edemas; Variações na pressão arterial e na frequência
cardíaca; Alterações periféricas.</p>
      <p>Além disso, é necessário obter informações precisas sobre cada um dos sintomas
referidos, como, por exemplo, verificar o início, o desencadeador da crise, o quanto as
atividades foram alteradas em virtude do sintoma (consegue realizar as atividades
domésticas? é possível realizar os cuidados de higiene?) e se foram realizadas atitudes
para o alívio (deitar-se, sentar-se, interromper a atividade iniciada).</p>
    </div>
  </div>
</div>
<script type="module" src=" ../backend/user_sus/loadDataUser.js"></script>
<script src=" ../js/activeButton/activeButtonGuiaEntrevista.js"></script>

```

```
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>
```

pages/pageHistoricoEnf.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link active" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link btn-logout" href="../../cadastro.html">
                    
                    <p class="text-menu">SAIR</p>

```

```

        </a>
      </li>
    </ul>
  </div>
</div>
</div>
</nav>
</div>
<div class="fundo-botao">
  <a href="/pageGuiaEntre.html" class="botao-padrao">
    <button>GUIA PARA ENTREVISTA</button>
  </a>
  <a href="/pageRecomend.html" class="botao-padrao">
    <button>RECOMENDAÇÕES PARA AVALIAÇÃO CLÍNICA</button>
  </a>
  <a href="/pageExamesLabExemplo.html" class="botao-padrao">
    <button>EXAMES LABORATORIAIS</button>
  </a>
  <a href="/pageAvaClinicExemplo.html" class="botao-padrao">
    <button>AVALIAÇÃO CLÍNICA E CÁLCULO DO RISCO CARDIOVASCULAR</button>
  </a>
</div>
</div>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pageImplement.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../css/global.css">
  <title>HAsupport</title>
</head>

<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">

```



```

<a class="nav-link" href="./adm_area.html">
  
  <p class="text-menu">MENU INICIAL</p>
</a>
</li>
<li class="nav-item">
  <a class="nav-link active" aria-current="page" href="./menu1.html">
    
    <p class="text-menu">GUIA PE</p>
  </a>
</li>
</li>
<li class="nav-item">
  <a class="nav-link" aria-current="page" href="./menu2.html">
    
    <p class="text-menu">REALIZACAO PE</p>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link btn-logout" href="./cadastro.html">
    
    <p class="text-menu">SAIR</p>
  </a>
</li>
</ul>
</div>
</div>
</div>
</div>
</nav>
</div>
<div class="text-style-pure">
  <p>Realização das ações e/ou intervenções estabelecidas no Planejamento de Enfermagem, as quais estão descritas sequencialmente após estabelecer os Diagnósticos de Enfermagem, na presente tecnologia. Ressalta-se a importância de considerar o contexto em que o(a) usuário(a) está inserido(a), bem como todos os dados compreendidos no decorrer do processo, além do suporte (ou não) da família e comunidade.</p>
</div>
</div>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
</body>

</html>

```

pages/pagePlanejEnfer.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../css/global.css">
  <title>HAsupport</title>
</head>

```

```

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="/adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="/menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="/cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </nav>
  </div>
  <div class="text-style-pure">
    <p>Etapa em que se determina os resultados esperados, bem como as ações e/ou intervenções
de enfermagem que serão realizadas conforme às respostas dos(as) usuários(as), família
ou comunidade em um certo momento do processo saúde e doença, essas foram
identificadas na etapa de Diagnóstico de Enfermagem. Como na presente tecnologia é
trabalhada a CIPE®, os resultados esperados bem como as intervenções já estão
apresentadas sequencialmente após estabelecer os Diagnósticos de Enfermagem.</p>
  </div>
</div>
<script src="/js/logoutSistema.js"></script>
<script src="/js/bootstrap-js/bootstrap.js"></script>
</body>

```

</html>

pages/pagePrintRelatorio.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

<link rel="stylesheet" href="../css/global.css">

<link rel="stylesheet" href="../css/displayNone.css">

<title>HAsupport</title>

</head>

<body>

<div class="fundo1">

<div class="menu">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="/adm\_area.html">



<p class="text-menu">MENU INICIAL</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu1.html">



<p class="text-menu">GUIA PE</p>

</a>

</li>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu2.html">



<p class="text-menu">REALIZACAO PE</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link btn-logout" href="/cadastro.html">



<p class="text-menu">SAIR</p>

</a>

</li>

</ul>

```

        </div>
    </div>
</div>
</nav>
</div>
<div class="fundo-botao fundo-botao-print">
    <!-- <div class="col-consultas">
        <div>
            <h2>Buscas</h2>
        </div>
        <form>
            <div>
                <input class="busca-usuarios" name="busca" required="required" type="text"
placeholder="Nome do usuário" />
            </div>
            <div>
                <a class="botao-padrao">
                    <button id="btn-print-ult-adic">BUSCAR</button>
                </a>
            </div>
        </form>
        <div class="retorno-select-1">
            < colocar aqui o retorno da escolha!! -->
        <!-- tabela temporaria!! -->
        <!-- <table>
            <tr>
                <th>Nome</th>
                <th>Idade</th>
                <th>Risco</th>
                <th>Data</th>
            </tr>
        </table>
        </div> -->
    <!-- </div> -->
<div class="col-relatorios">
    <div>
        <h2>Relatórios (PDF)</h2>
    </div>
    <div>
        <select class="form-select form-select-pdf" onsubmit="return false">
            <option selected>Selecione uma opção</option>
            <option value="1">Usuários(as) com maior risco</option>
            <option value="2">Últimos(as) usuários(as) cadastrados(as)</option>
        </select>
    </div>
    <div>
        <a class="botao-padrao">
            <button id="btn-print-maior-risco">BAIXAR</button>
        </a>
    </div>
</div>
</div>
</div>
<script type="module" src="../../backend/relatorio/relatorios.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>

```

</html>

pages/pageRecomend.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

<link rel="stylesheet" href="../css/global.css">

<link rel="stylesheet" href="../css/displayNone.css">

<title>HAsupport</title>

</head>

<body>

<div class="fundo1">

<div class="menu">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="/adm\_area.html">



<p class="text-menu">MENU INICIAL</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="/pageHistoricoEnf.html">



<p class="text-menu">GUIA PE</p>

</a>

</li>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu2.html">



<p class="text-menu">REALIZACAO PE</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link btn-logout" href="/cadastro.html">



<p class="text-menu">SAIR</p>

</a>

</li>

</ul>



<div class="text-style-pure2">

<p>É o processo de observação, em que a visão e o olfato da(o) profissional serão imprescindíveis, nesse sentido, é possível identificar dismorfias, distúrbios do desenvolvimento, lesões, secreções, presença de dispositivos e etc. Ainda, analisar a postura, modo de andar, formas de comunicação verbal e corporal, bem como é muito importante para o(a) usuário(a) com hipertensão arterial, avaliar a presença de dor e/ou desconforto. Ademais, a presença de diferentes odores, podendo indicar higienização precária, falta de produtos de higiene, por exemplo.</p>

</div>

</div>

<div>

<h4>Palpação</h4>

<div class="text-style-pure2">

<p>Permite a obtenção dos dados a partir do tato e da pressão, possibilitando a identificação de modificações de textura, espessura, consistência, sensibilidade, volume, dureza, flutuação, elasticidade e edema.</p>

</div>

</div>

<div>

<h4>Ausculta cardíaca</h4>

<div class="text-style-pure2">

<p>Utilizando o estetoscópio obtém-se ruídos considerados como normais ou patológicos. Além dos sons em si, destaca-se a importância de observar intensidade, tom, duração e qualidade. Inicialmente identifica-se os focos da ausculta cardíaca, apresentados na figura abaixo:</p>

</div>



<p>Figura 1. Ilustração da localização dos focos da ausculta cardíaca. Elaborada pelo Grupo CuidarTech.</p>

<div class="text-style-pure2">

<p>1º) Foco Aórtico: 2º espaço intercostal direito junto ao esterno;</p>

<p>2º) Foco Pulmonar: 2º espaço intercostal esquerdo junto ao esterno;</p>

<p>3º) Foco Tricúspide: 5º espaço intercostal borda esternal esquerda – base do apêndice xifóide;</p>

<p>4º) Foco Mitral: cruzamento do 5º espaço intercostal esquerdo (ictus cordis) com a linha hemiclavicular esquerda.</p>

<p>Nos focos da ausculta deve-se avaliar as bulhas cardíacas quanto a ritmicidade, fonese, quantidade e presença de sopro. Ausculta-se, principalmente, as duas bulhas cardíacas, as quais serão descritas a seguir:</p>

<p>- Primeira Bulha (B1): ligada ao fechamento das valvas mitral e tricúspide (valvas atrioventriculares). Marca o início da sístole (contração ventricular). É mais audível com o diafragma do estetoscópio colocado sobre o ápice do coração (foco mitral) e no foco tricúspide. A intensidade pode variar em certos distúrbios patológicos.</p>

<p>- Segunda Bulha (B2): relacionada com o fechamento das valvas pulmonar e aórtica (semilunares), é mais audível com o diafragma do estetoscópio colocado sobre a base do coração, no segundo espaço intercostal direito, na região paraesternal (foco aórtico).</p>

<p>Sinaliza o final da sístole e o início da diástole (enchimento ventricular).</p>

<p>As bulhas são consideradas rítmicas quando o examinador consegue ouvi-las no mesmo intervalo de tempo e arrítmicas quando os intervalos não são idênticos. A fonese está relacionada a quanto estão audíveis, o normal é que elas sejam normofonéticas, mas também podem estar hipofonéticas (volume mais baixo) ou hiperfonéticas (volume mais alto). Uma das alterações que pode ser encontrada é o sopro, a qual indica que uma ou mais valvas cardíacas não estão em funcionamento adequado. Podendo ser classificado como sistólico, quando o som anormal é auscultado entre a primeira e a segunda bulha, e/ou diastólico, quando o som anormal é auscultado após a segunda bulha.</p>

```
</p>
</div>
</div>
</div>
<a class="botao-padrao">
  <button id="btn-sinais">SINAIS VITAIS<span class="icon-expand"></span></button>
</a>
<div class="text-none-sinais text-none-style">
  <div>
    <h4>Pulso</h4>
    <div class="text-style-pure2">
      <p>É verificado utilizando-se a polpa dos dedos indicador e médio, por meio
da palpação de uma artéria, geralmente a artéria radial, durante um minuto, contando o
número de batimentos e verificando as características: intensidade (cheio ou filiforme),
ritmicidade (regular ou irregular) e simetria (iguais em ambos os membros).</p>
    </div>
  </div>
  <div>
    <h4>Frequência Cardíaca</h4>
    <div class="text-style-pure2">
      <p>Pode diferenciar-se do pulso devido as arritmias
cardíacas, pode ser verificado por meio da ausculta do pulso apical, no quinto espaço
intercostal esquerdo na linha hemiclavicular, com estetoscópio.</p>
    </div>
  </div>
  <div>
    <h4>Frequência Respiratória</h4>
    <div class="text-style-pure2">
      <p>Contagem dos movimentos respiratórios durante um
minuto, sem que o(a) usuário(a) tenha consciência, também devem ser verificados o ritmo,
profundidade e presença de desconforto respiratório.</p>
    </div>
  </div>
  <div>
    <h4>Temperatura</h4>
    <div class="text-style-pure2">
      <p>Utilizando o termômetro, pode ser realizada na cavidade oral,
retal, axilar ou pavilhão auricular. A mais comum no âmbito da Atenção Primária à Saúde
é a temperatura axilar, nesse sentido, é preciso tomar cuidado para que essa região esteja
isenta de umidade do suor, pois o resultado da leitura pode ser alterado. Por ser uma região
mais externa do corpo, o termômetro deve permanecer por um período mais longo (5 a 7
minutos).</p>
    </div>
  </div>
  <div>
    <h4>Pressão Arterial</h4>
    <div class="text-style-pure2">
      <p>Deve ser verificada, de preferência, nos membros superiores pelo uso de um estetoscópio
e um esfigmomanômetro de tamanho apropriado, a seguir será apresentado um quadro
com as dimensões do manguito de acordo com a circunferência do braço.
    </p>
    </div>
    <br><br>
    <!--
      
    -->
  </div>
</div>
```



```
<div class="text-style-pure2">
  <h6 class="sub-title">Preparo do(a) usuário(a) para a medida da pressão arterial e orientações
  iniciais:</h6>
  <p>- Explicar o procedimento para o(a) usuário(a) e orientar que não converse durante a
  aferição (dúvidas devem ser explicadas antes ou após o procedimento);
  <br>- Repouso de pelo menos 5 min em ambiente silencioso;
  <br>- Assegurar que o(a) usuário(a) não esteja de bexiga cheia;
  <br>- Assegurar que o(a) usuário(a) não praticou exercício físico há pelo menos 60 minutos;
  <br>- Verificar que não houve ingestão de bebidas alcoólicas, café e alimentos, bem como não
  ter fumado nos 30 minutos antes do procedimento;
  <br>- Manter as pernas descruzadas, pés apoiados no chão, dorso recostado na cadeira e
  relaxado;
  <br>- Remover roupas do braço no qual será colocado o manguito;
  <br>- Posicionar o braço na altura do coração (nível do ponto do esterno ou quarto espaço
  intercostal), apoiado, com a palma da mão voltada para cima e o cotovelo ligeiramente
  fletido;
  <br>- Na primeira verificação do(a) usuário(a), verificar a pressão arterial nos dois braços, de
  preferência simultaneamente, sendo possível detectar diferenças. Use o braço com o
  maior valor como referência para as próximas verificações;
  <br>- Devem ser realizadas três medidas da pressão arterial, com intervalo de 1 a 2 minutos;
  e medidas adicionais somente se as duas primeiras apresentarem diferenças > 10 mmHg.
  </p>
</div>
<div class="text-style-pure2">
  <h6 class="sub-title">Procedimento de medida da pressão arterial:</h6>
  <p>- Medir a circunferência do braço do(a) usuário(a);
  <br>- Selecionar o manguito de acordo com o tamanho adequado ao braço;
  <br>- Colocar o manguito sem deixar folgas, acima da fossa cubital, cerca de 2 a 3 cm;
  <br>- Centralizar o meio da parte compressiva do manguito sobre a artéria braquial;
  <br>- Estimar o nível da pressão sistólica pela palpação do pulso radial;
  <br>- Palpar a artéria braquial na fossa cubital e colocar a campânula do estetoscópio sem
  compressão excessiva;
  <br>- Inflar rapidamente até ultrapassar 20 a 30 mmHg do nível estimado da pressão sistólica,
  obtido pela palpação;
  <br>- Proceder a deflação lentamente (velocidade de 2 mmHg por segundo);
  <br>- Determinar a pressão sistólica na ausculta do primeiro som (fase I de korotkoff) e, em
  seguida, aumentar ligeiramente a velocidade de deflação;
  <br>- Determinar a pressão diastólica no desaparecimento do som (fase V de korotkoff);
  <br>- Auscultar cerca de 20 a 30 mmHg abaixo do nível do som, para confirmar seu
  desaparecimento e, depois, proceder à deflação rápida e completa;
  <br>- Se os batimentos persistirem até o nível zero, determinar a pressão diastólica no
  abafamento dos sons (fase IV de korotkoff) e anotar valores da sistólica/diastólica/zero;
  <br>- Esperar em torno de 1 a 2 minutos antes de novas medidas;
  <br>- Informar os valores de pressão arterial obtidos para o(a) usuário(a);
  <br>- Registrar os valores sem "arredondamentos" e o membro no qual foi aferida a pressão
  arterial.
  </p>
</div>
</div>
</div>
</div>
</div>
<script type="module" src="../../js/activeButton/activeButtonRecom.js"></script>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
</body>
```

</html>

pages/relatorioAltoRisco.html

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../css/bootstrap-css/bootstrap.css">

<link rel="stylesheet" href="../css/global.css">

<link rel="stylesheet" href="../css/relatorio.css">

<title>HAsupport</title>

</head>

<body>

<div class="fundo1">

<div class="menu">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<div>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="/adm\_area.html">



<p class="text-menu">MENU INICIAL</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu1.html">



<p class="text-menu">GUIA PE</p>

</a>

</li>

</li>

<li class="nav-item">

<a class="nav-link" aria-current="page" href="/menu2.html">



<p class="text-menu">REALIZACAO PE</p>

</a>

</li>

<li class="nav-item">

<a class="nav-link btn-logout" href="/cadastro.html">



<p class="text-menu">SAIR</p>

</a>

</li>

</ul>

</div>

```

        </div>
    </div>
</nav>
</div>

<div class="body-relatorio">
    <div class="conteudo_relatorio">
        <h2 class="titulo-pdf">Usuários(as) com ALTO RISCO</h2>
        <div class="retorno-select-2">
            <table>
                <tr>
                    <th>Cartão do SUS</th>
                    <th>CPF</th>
                    <th>Nome</th>
                    <th>Idade</th>
                </tr>
                <tr>
                    <td>111222233334444</td>
                    <td>11122233344</td>
                    <td>TestBot</td>
                    <td>01</td>
                </tr>
            </table>
        </div>
        <div class="botao-relatorio">
            <a class="botao-padrao">
                <button id="" onclick="downloadPDF()">IMPRIMIR</button>
            </a>
        </div>
    </div>
</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
<script type="module" src="../../backend/relatorio/relAltoRisco.js"></script>
<script src="../../js/Script imprimir.js"></script>
<script src="../../js/html2pdf.bundle.min.js"></script>
</body>

</html>

```

pages/relatorioConsulta.html

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="../../css/relatorio.css">
    <title>HAsupport</title>
</head>

```

```

<body>
<div class="fundo1">
  <div class="menu">
    <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <div>
            <ul class="navbar-nav">
              <li class="nav-item">
                <a class="nav-link" href="/adm_area.html">
                  
                  <p class="text-menu">MENU INICIAL</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="/menu1.html">
                  
                  <p class="text-menu">GUIA PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" aria-current="page" href="/menu2.html">
                  
                  <p class="text-menu">REALIZACAO PE</p>
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link btn-logout" href="/cadastro.html">
                  
                  <p class="text-menu">SAIR</p>
                </a>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </div>

    <div class="body-relatorio">
      <div class="conteudo_relatorio">
        <h2 class="titulo-pdf">Última Consulta do Usuário(a)</h2>
        <div class="retorno-select-3">
          <!-- AQUI -->
        </div>
      </div>
      <div class="botao-relatorio">
        <a class="botao-padrao">
          <button id="" onclick="downloadPDF()">IMPRIMIR</button>
        </a>
      </div>
    </div>

```

```
</div>
</div>
<script src="../../js/logoutSistema.js"></script>
<script src="../../js/bootstrap-js/bootstrap.js"></script>
<script type="module" src="../../backend/relatorio/relConsulta.js"></script>
<script src="../../js/Script imprimir.js"></script>
<script src="../../js/html2pdf.bundle.min.js"></script>
</body>
```

```
</html>
```

pages/relatorioTodosUsers.html

```
<!DOCTYPE html>
<html lang="pt-br">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../../css/bootstrap-css/bootstrap.css">
  <link rel="stylesheet" href="../../css/global.css">
  <link rel="stylesheet" href="../../css/relatorio.css">
  <title>HAsupport</title>
</head>
```

```
<body>
  <div class="fundo1">
    <div class="menu">
      <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarNav">
            <div>
              <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="../../adm_area.html">
                    
                    <p class="text-menu">MENU INICIAL</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu1.html">
                    
                    <p class="text-menu">GUIA PE</p>
                  </a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="../../menu2.html">
                    
                    <p class="text-menu">REALIZACAO PE</p>
                  </a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </div>
</body>
```

```

        </li>
        <li class="nav-item">
            <a class="nav-link btn-logout" href="/cadastro.html">
                
                <p class="text-menu">SAIR</p>
            </a>
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="body-relatorio">
    <div class="conteudo_relatorio">
        <h2 class="titulo-pdf">Usuários(as) com ALTO RISCO</h2>
        <div class="retorno-select-2">
            <table>
                <tr>
                    <th>Cartão do SUS</th>
                    <th>CPF</th>
                    <th>Nome</th>
                    <th>Idade</th>
                </tr>
                <tr>
                    <td>111222233334444</td>
                    <td>11122233344</td>
                    <td>TestBot</td>
                    <td>01</td>
                </tr>
            </table>
        </div>
    </div>
    <div class="botao-relatorio">
        <a class="botao-padrao">
            <button id="" onclick="downloadPDF()">IMPRIMIR</button>
        </a>
    </div>
</div>
</div>
<script src="../js/logoutSistema.js"></script>
<script src="../js/bootstrap-js/bootstrap.js"></script>
<script type="module" src="../backend/relatorio/relTodosUsers.js"></script>
<script src="../js/Script imprimir.js"></script>
<script src="../js/html2pdf.bundle.min.js"></script>

</body>

</html>

<!--teste-->

```

main.js

```

// Modules to control application life and create native browser window
const {app, BrowserWindow} = require('electron')
const path = require('path')

```

```

function createWindow () {
  // Create the browser window.
  const mainWindow = new BrowserWindow({
    width: 1280,
    height: 1024,
    webPreferences: {
      nodeIntegration: true,
      contextIsolation: false,
      worldSafeExecuteJavaScript: true,
      enableRemoteModule: false,
      preload: path.join(__dirname, 'preload.js')
    },
    icon: __dirname+'./assets/enf2.png',
  })

  //Comentar linha abaixo pra aparecer menu na janela do software
  mainWindow.setMenuBarVisibility(false);

  // and load the index.html of the app.
  mainWindow.loadFile('./pages/cadastro.html')

  // Open the DevTools.
  // mainWindow.webContents.openDevTools()
}

// This method will be called when Electron has finished
// initialization and is ready to create browser windows.
// Some APIs can only be used after this event occurs.
app.whenReady().then(() => {
  createWindow()

  app.on('activate', function () {
    // On macOS it's common to re-create a window in the app when the
    // dock icon is clicked and there are no other windows open.
    if (BrowserWindow.getAllWindows().length === 0) createWindow()
  })
})

// Quit when all windows are closed, except on macOS. There, it's common
// for applications and their menu bar to stay active until the user quits
// explicitly with Cmd + Q.
app.on('window-all-closed', function () {
  if (process.platform !== 'darwin') app.quit()
})

// In this file you can include the rest of your app's specific main process
// code. You can also put them in separate files and require them here.

```