

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO 1 - Taller 2.5

DOCUMENTACIÓN



Rodríguez Pintos, Mateo Santiago – 192391



García da Rosa Texo, Thiago 317133

Grupo N2.5 E

Docente: Sergio Palay

Analista en Tec. de la Información

Fecha de entrega del documento 22-02-2024

Índice:

1. Código Fuente	3
2. Capturas de pantalla APP	28

Github: https://github.com/MateoRodriguezzi/Obligatorio_Taller_Dis_Moviles

HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Obligatorio</title>

  <script type="module"
src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/ionic.esm.js"></script>
  <script nomodule
src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/ionic.js"></script>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/@ionic/core/css/ionic.bundle.css" />
  <link rel="stylesheet" href="css/style.css">

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
    integrity="sha256-p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY=" crossorigin=""
  />
  <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"
    integrity="sha256-20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV11vTlZBo="
crossorigin=""></script>

</head>

<body>
  <ion-app>

    <!-- RUTEO ASEGURA LA NAVEGABILIDAD -->

    <ion-router id="ruteo">
      <ion-route url="/" component="page-home"> </ion-route>
      <ion-route url="/login" component="page-login"> </ion-route>
    </ion-router>
  </ion-app>
</body>
</html>
```

```

    <ion-route url="/registrarU" component="page-registrar"> </ion-route>
    <ion-route url="/registrarC" component="page-registrar-comida">
</ion-route>
    <ion-route url="/listar" component="page-listar-comida"> </ion-route>
    <ion-route url="/informeC" component="page-informe-calorias"> </ion-route>
    <ion-route url="/mapa" component="page-mapa"> </ion-route>
</ion-router>
<ion-nav></ion-nav>

<!-- MENU DESPLEGABLE -->

<ion-menu content-id="main-content" id="menu">
  <ion-header>
    <ion-toolbar>
      <ion-title>Selecciona una opción</ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-content class="ion-padding">

    <ion-list lines="full">

      <ion-item button id="btnHome" href="/" onclick="cerrarMenu()">
        <ion-icon slot="start" name="home-outline"></ion-icon>
        <ion-label>Home</ion-label>
      </ion-item>

      <ion-item button id="btnLogin" href="/login"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="log-in-outline"></ion-icon>
        <ion-label>Login</ion-label>
      </ion-item>

      <ion-item button id="btnRegistrarU" href="/registrarU"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="person-add-outline"></ion-icon>
        <ion-label>Registrar Usuario</ion-label>
      </ion-item>

      <ion-item button id="btnRegistrarC" href="/registrarC"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="add-outline"></ion-icon>

```

```

        <ion-label>Registrar Comida</ion-label>
      </ion-item>

      <ion-item button id="btnListar" href="/listar"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="list-outline"></ion-icon>
        <ion-label>Listar Comidas</ion-label>
      </ion-item>

      <ion-item button id="btnInformar" href="/informeC"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="stats-chart-outline"></ion-icon>
        <ion-label>Informe de Calorías</ion-label>
      </ion-item>

      <ion-item button id="btnMapa" href="/mapa" onclick="cerrarMenu()">
        <ion-icon slot="start" name="map-outline"></ion-icon>
        <ion-label>Mapa</ion-label>
      </ion-item>

      <ion-item button id="btnCerrarSesion" href="/login"
onclick="cerrarMenu()">
        <ion-icon slot="start" name="log-out-outline"></ion-icon>
        <ion-label>Logout</ion-label>
      </ion-item>

    </ion-list>

  </ion-content>
</ion-menu>

<div class="ion-page" id="main-content">

</div>

<!-- PANTALLAS -->

<page-home class="ion-page" id="pantalla-home">
  <ion-header>
    <ion-toolbar>
      <ion-title>Bienvenido</ion-title>
      <ion-buttons slot="start">

```

```

        <ion-menu-button></ion-menu-button>

    </ion-buttons>

</ion-toolbar>

</ion-header>

<ion-content class="ion-padding" fullscreen>
    <p id="mensajeLoginExitoso" style="color: green;"></p>
    <ion-img src="./img/ejercicio.jpg" alt="Persona activa y
saludable"></ion-img>
    <p class="custom-text">Empezá hoy a escribir el capítulo más saludable
de tu vida</p>
    <p id="mensajeLogueo"></p>
</ion-content>
</page-home>

<page-login class=" ion-page" id="pantalla-login">
    <ion-header>
        <ion-toolbar>
            <ion-title>Login</ion-title>
            <ion-buttons slot="start">
                <ion-menu-button></ion-menu-button>
            </ion-buttons>
        </ion-toolbar>
    </ion-header>
    <ion-content class="ion-padding" fullscreen>

        <ion-item>
            <ion-label position="floating">Usuario</ion-label>
            <ion-input type="email" placeholder="Ingrese su usuario"
id="txtLoginUsuario"></ion-input>
        </ion-item>

        <ion-item>
            <ion-label position="floating">Contraseña</ion-label>
            <ion-input type="password" placeholder="Ingrese su contraseña"
id="txtPasswordUsuario"></ion-input>
        </ion-item>

        <p id="mensajeError" style="color: red;"></p>
    
```

```

<ion-button expand="block" id="btnHacerLogin">Iniciar
sesión</ion-button>
</ion-content>
</page-login>

<page-registrar class="ion-page" id="pantalla-registroU">
  <ion-header>
    <ion-toolbar>
      <ion-title>Registro en la aplicación</ion-title>
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
    </ion-toolbar>
  </ion-header>
  <ion-content class="ion-padding" fullscreen>
    <ion-item>
      <ion-label position="floating">Usuario</ion-label>
      <ion-input type="text" name="usuario" id="txtRegistroUsuario"
required></ion-input>
    </ion-item>
    <ion-item>
      <ion-label position="floating">Contraseña</ion-label>
      <ion-input type="password" name="contrasena"
id="txtRegistroPassword" required></ion-input>
    </ion-item>
    <ion-item>
      <ion-label position="floating">País de residencia</ion-label>
      <ion-select type="text" name="pais" id="txtRegistroPais"
required></ion-select>
    </ion-item>
    <ion-item>
      <ion-label position="floating">Calorías diarias</ion-label>
      <ion-input type="number" name="calorias" id="txtRegistroCalorias"
required></ion-input>
    </ion-item>
    <ion-button expand="block" id="btnRegistrarUsuario">Registrar
Usuario</ion-button>
    <p id="mensajeRegistroExitoso" style="color: green;"></p>
    <p id="mensajeErrorRegistro" style="color: red;"></p>
  </ion-content>
</page-registrar>

```

```
<page-registrar-comida class="ion-page" id="pantalla-registroC">
  <ion-header>
    <ion-toolbar>
      <ion-title>Registrar Comida</ion-title>
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
    </ion-toolbar>
  </ion-header>
  <ion-content class="ion-padding" fullscreen>
    <ion-item>
      <ion-label position="floating">Seleccione alimento</ion-label>
      <ion-select name="pais" id="txtIdAlimento" required></ion-select>
    </ion-item>
    <ion-item>
      <ion-label position="floating">Cantidad</ion-label>
      <ion-input type="number" name="txtCantidadAlimento"
id="txtCantidadAlimento" required></ion-input>
    </ion-item>
    <ion-item>
      <ion-label position="floating"></ion-label>
      <ion-input type="date" name="fecha" id="txtFechaAlimento"
min="1990-01-01"
      max="{{ getToday() | date: 'yyyy-MM-dd' }}"
display-format="DD/MM/YYYY" required></ion-input>
    </ion-item>
    <script>
      var inputFechaAlimento =
document.getElementById('txtFechaAlimento');
      var fechaActual = new Date().toISOString().split('T')[0];
      inputFechaAlimento.max = fechaActual;
    </script>
    <ion-button expand="block" id="btnRegistrarAlimento">Registrar
Alimento</ion-button>
    <p id="mensajeRegistroAlimentoExitoso" style="color: green;"></p>
    <p id="mensajeErrorAlimentoRegistro" style="color: red;"></p>
  </ion-content>
</page-registrar-comida>

<page-listar-comidas class="ion-page" id="pantalla-listarC">
```



```

<ion-header>
  <ion-toolbar>
    <ion-title>Listado de Comidas</ion-title>
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
<ion-content class="ion-padding" fullscreen>
  <ion-list id="lista-comidas">
    <!-- Aquí se mostrarán los registros -->
  </ion-list>
  <p id="mensajeSinRegistros" style="color: red; display: none;
text-align: center;">No tienes registros.
  </p>
  <div style="margin-top: 20px;">
    <ion-item>
      <ion-label position="stacked" style="margin-bottom:
5px;">Desde:</ion-label>
      <ion-input type="date" id="fechaInicio"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label position="stacked" style="margin-bottom:
5px;">Hasta:</ion-label>
      <ion-input type="date" id="fechaFin"></ion-input>
    </ion-item>
    <ion-button expand="block" id="btnFiltrar" style="margin-top:
20px;">Filtrar</ion-button>
  </div>
</ion-content>
</page-listar-comidas>

<page-informe-calorias class="ion-page" id="pantalla-informeCalorias">
  <ion-header>
    <ion-toolbar>
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title>Informe de Calorías</ion-title>
    </ion-toolbar>
  </ion-header>

```

```

<ion-content class="ion-padding" fullscreen>
  <div class="calorias-container">
    <div class="calorias-card">
      <ion-icon name="flame-outline"
class="calorias-icon"></ion-icon>
      <ion-label> Total Calorías Ingeridas:</ion-label>
      <ion-label id="totalCalorias"
class="calorias-value"></ion-label>
    </div>
    <div class="calorias-card">
      <ion-icon name="today-outline"
class="calorias-icon"></ion-icon>
      <ion-label> Calorías Registradas Hoy:</ion-label>
      <ion-label id="caloriasDiarias"
class="calorias-value"></ion-label>
    </div>
  </div>
</ion-content>
</page-informe-calorias>

<page-mapa class="ion-page" id="pantalla-mapa">
  <ion-header>
    <ion-toolbar>
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title>Mapa registros de usuarios</ion-title>
    </ion-toolbar>
  </ion-header>

  <ion-content>
    <ion-item lines="none">
      <ion-label position="floating">Cantidad de usuarios</ion-label>
      <ion-input placeholder="Ingrese la cantidad de usuarios"
type="number"
      id="userRecordsInput"></ion-input>
      <ion-button id="searchButton" color="primary">Buscar</ion-button>
      <p id="mensajeErrorMapa" style="color: red;"></p>
    </ion-item>
    <div id="map"></div>
  </ion-content>

```

```
</page-mapa>

</ion-app>
<script src="js/codigo.js"></script>
</body>

</html>
```

JS:

```
document
  .getElementById("btnFiltrar")
  .addEventListener("click", filtrarRegistrosPorFecha);

const MENU = document.querySelector("#menu");
const ROUTER = document.querySelector("#ruteo");
const HOME = document.querySelector("#pantalla-home");
const LOGIN = document.querySelector("#pantalla-login");
const REGISTROU = document.querySelector("#pantalla-registroU");
const REGISTROC = document.querySelector("#pantalla-registroC");
const LISTADO = document.querySelector("#pantalla-listarC");
const INFORME = document.querySelector("#pantalla-informeCalorias");
const MAPA = document.querySelector("#pantalla-mapa");
const URLBASE = "https://calcount.develotion.com/";

let registrosArray = [];
let listaAlimentos = [];
let paises = [];
let arrayUsuariosPorPais = [];

let map;
let mapaInicializado = false;
let miLatitud;
let miLongitud;
let popup;
let marker;
```

```
inicio();

function cerrarMenu() {
    MENU.close();
}

function inicio() {
    chequearSession();
    eventos();
    obtenerPaíses();
    obtenerAlimentos();
    usuariosPorPaís();
    armarMapa();
}

function eventos() {
    ROUTER.addEventListener("ionRouteDidChange", navegar);
    document
        .querySelector("#btnHacerLogin")
        .addEventListener("click", previaLogin);
    document
        .querySelector("#btnRegistrarUsuario")
        .addEventListener("click", previaRegistro);
    document
        .querySelector("#btnCerrarSesion")
        .addEventListener("click", cerrarSesion);
    document
        .querySelector("#btnRegistrarC")
        .addEventListener("click", obtenerAlimentos);
    document
        .querySelector("#btnRegistrarAlimento")
        .addEventListener("click", previaRegistroAlimento);
    document
        .querySelector("#btnInformar")
        .addEventListener("click", calcularCalorias);
    document
        .querySelector("#btnInformar")
        .addEventListener("click", obtenerAlimentos);
    document
        .querySelector("#searchButton")
        .addEventListener("click", paísesConMasUsuarios);
}
```

```
function navegar(evt) {
    ocultarPantallas();
    if (evt.detail.to == "/") HOME.style.display = "block";
    if (evt.detail.to == "/login") LOGIN.style.display = "block";
    if (evt.detail.to == "/registrarU") REGISTROU.style.display = "block";
    if (evt.detail.to == "/registrarC") REGISTROC.style.display = "block";
    if (evt.detail.to == "/listar") {
        LISTADO.style.display = "block";
        obtenerRegistros();
    }
    if (evt.detail.to == "/informeC") INFORME.style.display = "block";
    if (evt.detail.to == "/mapa") MAPA.style.display = "block";
}

function ocultarPantallas() {
    HOME.style.display = "none";
    LOGIN.style.display = "none";
    REGISTROU.style.display = "none";
    REGISTROC.style.display = "none";
    LISTADO.style.display = "none";
    INFORME.style.display = "none";
    MAPA.style.display = "none";
}

function ocultarBotones() {
    document.querySelector("#btnHome").style.display = "none";
    document.querySelector("#btnLogin").style.display = "none";
    document.querySelector("#btnRegistrarU").style.display = "none";
    document.querySelector("#btnRegistrarC").style.display = "none";
    document.querySelector("#btnListar").style.display = "none";
    document.querySelector("#btnInformar").style.display = "none";
    document.querySelector("#btnMapa").style.display = "none";
    document.querySelector("#btnCerrarSesion").style.display = "none";
}

function chequearSession() {
    if (localStorage.getItem("user") != null) {
        MostrarMenuActivo();
        getMiPosicion();
    } else {
        MostrarMenu();
    }
}
```

```
}  
}  
  
function MostrarMenuActivo() {  
    ocultarBotones();  
    document.querySelector("#btnHome").style.display = "block";  
    document.querySelector("#btnRegistrarC").style.display = "block";  
    document.querySelector("#btnListar").style.display = "block";  
    document.querySelector("#btnInformar").style.display = "block";  
    document.querySelector("#btnMapa").style.display = "block";  
    document.querySelector("#btnCerrarSesion").style.display = "block";  
}  
  
function MostrarMenu() {  
    ocultarBotones();  
    document.querySelector("#btnLogin").style.display = "block";  
    document.querySelector("#btnRegistrarU").style.display = "block";  
}  
  
function previaLogin() {  
    let usuario = document.querySelector("#txtLoginUsuario").value;  
    let password = document.querySelector("#txtPasswordUsuario").value;  
  
    let unUsuario = new Object();  
    unUsuario.usuario = usuario;  
    unUsuario.password = password;  
  
    hacerLogin(unUsuario);  
}  
  
function hacerLogin(unUsuario) {  
    fetch(`${URLBASE}login.php`, {  
        method: "POST",  
        headers: {  
            "Content-Type": "application/json",  
        },  
        body: JSON.stringify(unUsuario),  
    })  
    .then(function (response) {  
        console.log(response);  
        return response.json();  
    })  
}
```

```
.then(function (data) {
  console.log(data);
  if (data.codigo === 200) {
    document.getElementById("mensajeLoginExitoso").innerText =
      "Login exitoso!";
    localStorage.setItem("idUsuario", data.id);
    localStorage.setItem("apikey", data.apiKey);
    localStorage.setItem("caloriasDiarias", data.caloriasDiarias);
    ocultarPantallas();
    HOME.style.display = "block";
    MostrarMenuActivo();
    document.querySelector("#pantalla-home ion-title").innerText =
      "Bienvenido " + unUsuario.usuario;
    setTimeout(function () {
      document.getElementById("mensajeLoginExitoso").innerText = "";
    }, 4000);
  } else {
    document.getElementById("mensajeError").innerText = data.mensaje;
  }
})
.catch(function (error) {
  console.log(error);
});
});

function previaRegistro() {
  let usuario = document.querySelector("#txtRegistroUsuario").value;
  let password = document.querySelector("#txtRegistroPassword").value;
  let idPais = document.querySelector("#txtRegistroPais").value;
  let caloriasDiarias = document.querySelector("#txtRegistroCalorias").value;

  let nuevoUsuario = new Object();
  nuevoUsuario.usuario = usuario;
  nuevoUsuario.password = password;
  nuevoUsuario.idPais = idPais;
  nuevoUsuario.caloriasDiarias = caloriasDiarias;

  hacerRegistro(nuevoUsuario);
}

function hacerRegistro(nuevoUsuario) {
  fetch(`${URLBASE}usuarios.php`, {
```

```
method: "POST",
headers: {
  "Content-Type": "application/json",
},
body: JSON.stringify(nuevoUsuario),
})

.then(function (response) {
  console.log(response);
  return response.json();
})

.then(function (data) {
  if (data.codigo === 200) {
    document.getElementById("mensajeRegistroExitoso").innerText =
      "Registro exitoso!";
    document.getElementById("mensajeErrorRegistro").innerText = "";
    limpiarCamposRegistro();
  } else {
    document.getElementById("mensajeErrorRegistro").innerText =
      data.mensaje;
  }
})

.catch(function (error) {
  console.log(error);
});
});

function limpiarCamposRegistro() {
  document.querySelector("#txtRegistroUsuario").value = "";
  document.querySelector("#txtRegistroPassword").value = "";
  document.querySelector("#txtRegistroPais").value = "";
  document.querySelector("#txtRegistroCalorias").value = "";
}

function limpiarCamposRegistroAlimento() {
  document.querySelector("#txtIdAlimento").value = "";
  document.querySelector("#txtCantidadAlimento").value = "";
  document.querySelector("#txtFechaAlimento").value = "";
}

function cerrarSesion() {
  localStorage.clear();
  window.location.reload();
}
```



```
ROUTER.push("/login");
}

function obtenerPaíses() {
  fetch(`${URLBASE}países.php`, {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then(function (response) {
    return response.json();
  })
  .then(function (respuesta) {
    console.log(respuesta);
    if (respuesta && respuesta.países) {
      países = respuesta.países;
      const selectPais = document.getElementById("txtRegistroPais");
      respuesta.países.forEach(function (pais) {
        const option = document.createElement("ion-select-option");
        option.value = pais.id;
        option.textContent = pais.name;
        selectPais.appendChild(option);
      });
    } else {
      console.error("Error al obtener los países: Datos incompletos");
    }
  })
  .catch(function (error) {
    console.error("Error al obtener los países:", error);
  });
}

function obtenerAlimentos() {
  if (listaAlimentos.length === 0) {
    apiKey = localStorage.getItem("apikey");
    idUser = localStorage.getItem("idUser");

    fetch(`${URLBASE}alimentos.php`, {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
      },
    })
    .then(function (response) {
      return response.json();
    })
    .then(function (respuesta) {
      console.log(respuesta);
      if (respuesta && respuesta.alimentos) {
        alimentos = respuesta.alimentos;
        const selectAlimento = document.getElementById("txtRegistroAlimento");
        respuesta.alimentos.forEach(function (alimento) {
          const option = document.createElement("ion-select-option");
          option.value = alimento.id;
          option.textContent = alimento.name;
          selectAlimento.appendChild(option);
        });
      } else {
        console.error("Error al obtener los alimentos: Datos incompletos");
      }
    })
    .catch(function (error) {
      console.error("Error al obtener los alimentos:", error);
    });
  }
}
```

```

    apiKey: apiKey,
    iduser: idUser,
  },
})
.then(function (response) {
  return response.json();
})
.then(function (respuesta) {
  if (respuesta && respuesta.alimentos) {
    const selectAlimento = document.getElementById("txtIdAlimento");
    respuesta.alimentos.forEach(function (alimento) {
      const option = document.createElement("ion-select-option");
      option.value = alimento.id;
      option.textContent = alimento.nombre;
      selectAlimento.appendChild(option);
    });
    listaAlimentos = respuesta.alimentos;
  } else {
    console.error("Error al obtener los alimentos");
  }
})
.catch(function (error) {
  console.error("Error al obtener los alimentos:", error);
});
}
}

function previaRegistroAlimento() {
  let idUsuario = localStorage.getItem("idUsuario");
  let id = document.querySelector("#txtIdAlimento").value;
  let cantidad = document.querySelector("#txtCantidadAlimento").value;
  let fecha = document.querySelector("#txtFechaAlimento").value;

  let nuevoAlimentoRegistrado = new Object();
  nuevoAlimentoRegistrado.idAlimento = id;
  nuevoAlimentoRegistrado.idUsuario = idUsuario;
  nuevoAlimentoRegistrado.cantidad = cantidad;
  nuevoAlimentoRegistrado.fecha = fecha;

  hacerRegistroAlimento(nuevoAlimentoRegistrado);
}

```

```
function hacerRegistroAlimento(nuevoAlimentoRegistrado) {
  apiKey = localStorage.getItem("apikey");
  idUser = localStorage.getItem("idUser");

  fetch(`${URLBASE}registros.php`, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      apikey: apiKey,
      iduser: idUser,
    },
    body: JSON.stringify(nuevoAlimentoRegistrado),
  })
  .then(function (response) {
    console.log(response);
    return response.json();
  })
  .then(function (data) {
    console.log(data);
    if (data.codigo === 200) {
      localStorage.setItem("idRegistro", data.idRegistro);
      document.getElementById("mensajeRegistroAlimentoExitoso").innerText =
        "Registro exitoso!";
      document.getElementById("mensajeErrorAlimentoRegistro").innerText = "";
      limpiarCamposRegistroAlimento();
    } else {
      document.getElementById("mensajeErrorAlimentoRegistro").innerText =
        data.mensaje;
    }
  })
  .catch(function (error) {
    console.log(error);
  });
}

function obtenerRegistros() {
  const apiKey = localStorage.getItem("apikey");
  const idUsuario = localStorage.getItem("idUser");

  fetch(`${URLBASE}registros.php?idUsuario=${idUsuario}`, {
    method: "GET",
    headers: {
```

```
"Content-Type": "application/json",
apikey: apiKey,
iduser: idUsuario,
},
}))
.then(function (response) {
    return response.json();
})
.then(function (data) {
    console.log(data);
    if (data && data.registros) {
        registrosArray = data.registros;
        mostrarRegistros(registrosArray);
    } else {
        console.error("Error al obtener los registros");
    }
})
.catch(function (error) {
    console.error("Error al obtener los registros:", error);
});
});

function mostrarRegistros(registros) {
    const listaComidas = document.getElementById("lista-comidas");
    listaComidas.innerHTML = "";

    if (registros.length === 0) {
        document.getElementById("mensajeSinRegistros").style.display = "block";
        return;
    } else {
        document.getElementById("mensajeSinRegistros").style.display = "none";
    }

    registros.forEach(function (registro) {
        const item = document.createElement("ion-item");

        let alimentoEncontrado = null;
        for (let i = 0; i < listaAlimentos.length; i++) {
            if (listaAlimentos[i].id === registro.idAlimento) {
                alimentoEncontrado = listaAlimentos[i];
                break;
            }
        }
    });
}
```

```

}

if (alimentoEncontrado) {
  const imagen = document.createElement("ion-img");
  imagen.src = `${URLBASE}imgs/${alimentoEncontrado.imagen}.png`;
  imagen.slot = "start";
  item.appendChild(imagen);

  const texto = document.createElement("ion-label");
  texto.textContent = `${alimentoEncontrado.nombre} -
${alimentoEncontrado.calorias} calorías`;
  item.appendChild(texto);
} else {
  console.error(`No se encontró el alimento con id ${registro.idAlimento}`);
}

const botonEliminar = document.createElement("ion-button");
botonEliminar.slot = "end";
botonEliminar.textContent = "Eliminar";
botonEliminar.addEventListener("click", function () {
  eliminarRegistro(registro.id);
});
item.appendChild(botonEliminar);
listaComidas.appendChild(item);
});
}

function eliminarRegistro(idRegistro) {
  const apiKey = localStorage.getItem("apikey");
  const idUsuario = localStorage.getItem("idUsuario");

  fetch(`${URLBASE}registros.php?idRegistro=${idRegistro}`, {
    method: "DELETE",
    headers: {
      "Content-Type": "application/json",
      apikey: apiKey,
      iduser: idUsuario,
    },
  })
  .then(function (response) {
    if (response.ok) {
      mostrarMensajeTemporal("Registro eliminado con éxito", "red");
    }
  })
}

```

```

        obtenerRegistros();
    } else {
        mostrarMensajeTemporal("Error al eliminar el registro");
    }
})
.catch(function (error) {
    console.error("Error al eliminar el registro:", error);
});
}

function mostrarMensajeTemporal(mensaje, color) {
    const mensajeElemento = document.createElement("div");
    mensajeElemento.textContent = mensaje;
    mensajeElemento.classList.add("mensaje-temporal");
    mensajeElemento.style.backgroundColor = color;

    document.body.appendChild(mensajeElemento);

    setTimeout(function () {
        mensajeElemento.classList.add("mostrar");
    }, 100);

    setTimeout(function () {
        mensajeElemento.classList.remove("mostrar");
        setTimeout(function () {
            document.body.removeChild(mensajeElemento);
        }, 300);
    }, 2000);
}

function filtrarRegistrosPorFecha() {
    const fechaInicio = new Date(document.getElementById("fechaInicio").value);
    const fechaFin = new Date(document.getElementById("fechaFin").value);

    const registrosFiltrados = [];

    for (let i = 0; i < registrosArray.length; i++) {
        const fechaRegistro = new Date(registrosArray[i].fecha);
        if (fechaRegistro >= fechaInicio && fechaRegistro <= fechaFin) {
            registrosFiltrados.push(registrosArray[i]);
        }
    }
}

```

```
mostrarRegistros(registrosFiltrados);
}

function calcularCalorias() {
  let totalCalorias = 0;
  let caloriasHoy = 0;
  const hoy = new Date().toISOString().split("T")[0];

  for (let i = 0; i < registrosArray.length; i++) {
    for (let j = 0; j < listaAlimentos.length; j++) {
      if (registrosArray[i].idAlimento === listaAlimentos[j].id) {
        const cantidad = registrosArray[i].cantidad;
        const caloriasPorAlimento = listaAlimentos[j].calorias * cantidad;
        totalCalorias += caloriasPorAlimento;

        if (registrosArray[i].fecha === hoy) {
          caloriasHoy += caloriasPorAlimento;
        }
        break;
      }
    }
  }

  document.getElementById("totalCalorias").textContent = totalCalorias;
  document.getElementById("caloriasDiarias").textContent = caloriasHoy;

  const caloriasDiariasPrevistas = parseInt(
    localStorage.getItem("caloriasDiarias")
  );

  const porcentaje = (caloriasHoy / caloriasDiariasPrevistas) * 100;
  let colorTexto = "";
  if (porcentaje > 100) {
    colorTexto = "red";
  } else if (porcentaje >= 90 && porcentaje <= 100) {
    colorTexto = "orange";
  } else {
    colorTexto = "green";
  }

  document.getElementById("caloriasDiarias").style.color = colorTexto;
}
```

```

}

function armarMapa() {
  if (!mapaInicializado) {
    function miUbicacion(position) {
      miLatitud = position.coords.latitude;
      miLongitud = position.coords.longitude;

      if (map) {
        map.remove();
      }

      map = L.map("map").setView([miLatitud, miLongitud], 16);

      L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
        maxZoom: 19,
        attribution:
          '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>',
      }).addTo(map);

      L.marker([miLatitud, miLongitud])
        .addTo(map)
        .bindPopup("Usted está aquí.")
        .openPopup();
    }

    function errorEnGeo(error) {
      console.log("Error en la geolocalización:", error.message);
    }

    navigator.geolocation.getCurrentPosition(miUbicacion, errorEnGeo);

    mapaInicializado = true;
  }
}

function getMiPosicion() {
  navigator.geolocation.getCurrentPosition(miUbicacion, errorEnGeo);
}

function usuariosPorPais() {
  const apiKey = localStorage.getItem("apikey");

```



```
const idUsuario = localStorage.getItem("idUsuario");

const headers = {
  "Content-Type": "application/json",
  apikey: apiKey,
  iduser: idUsuario,
};

fetch(`${URLBASE}usuariosPorPais.php`, {
  method: "GET",
  headers: headers,
})
  .then(function (response) {
    if (!response.ok) {
      throw new Error("No se pudo obtener la información.");
    }
    return response.json();
  })
  .then(function (data) {
    arrayUsuariosPorPais = data.paises;
  })
  .catch(function (error) {
    console.error("Error al obtener los datos:", error);
  });
}

function paisesConMasUsuarios() {
  const cantidadUsuarios = parseInt(
    document.querySelector("#userRecordsInput").value
  );
  console.log("Cantidad de usuarios ingresada:", cantidadUsuarios);

  if (isNaN(cantidadUsuarios) || cantidadUsuarios < 0) {
    document.getElementById("mensajeErrorMapa").innerText =
      "Debes ingresar un valor válido";
    return;
  } else {
    document.getElementById("mensajeErrorMapa").innerText = ""; // Borra el mensaje de
error
  }

  arrayUsuariosPorPais.forEach(function (paisUsuario) {
```

```
if (paisUsuario.cantidadDeUsuarios > cantidadUsuarios) {  
  const paisInfo = paises.find((pais) => pais.id === paisUsuario.id);  
  if (paisInfo) {  
    agregarMarcador(paisInfo);  
  }  
}  
});  
}  
  
function agregarMarcador(pais) {  
  console.log("Pais recibido:", pais);  
  L.marker([pais.latitude, pais.longitude])  
    .addTo(map)  
    .bindPopup(  
      `${pais.name}: <em>País con más registros que el ingresado</em>`  
    );  
}
```

CSS:

```
ion-menu {  
  --ion-background-color: #ffffff;  
  --ion-text-color: #282c34;  
}  
  
ion-menu ion-content {  
  background-color: #ffffff;  
  color: #282c34;  
}  
  
ion-toolbar {  
  --background: #282c34;  
  --ion-color-base: #ffffff;  
}  
  
ion-title {  
  color: #ffffff;  
}  
  
ion-content {  
  --background: #f0f2f5;
```

```
--color: #282c34;
}

ion-menu-button {
  --color: #ffffff;
  --background: transparent;
}

ion-menu-button::part(icon) {
  font-size: 1.5rem;
}

.custom-title {
  font-size: 24px;
  font-weight: bold;
  color: #ffffff;
}

.custom-text {
  font-size: 18px;
  color: #333333;
}

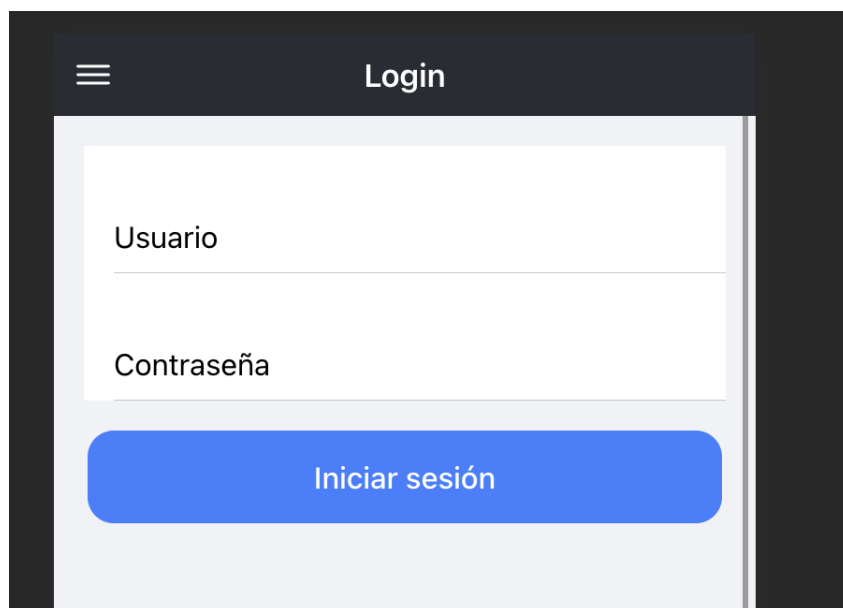
.mensaje-temporal {
  position: fixed;
  top: 20px;
  left: 50%;
  transform: translateX(-50%);
  color: white;
  padding: 10px 20px;
  border-radius: 5px;
  font-size: 16px;
  font-family: Arial, sans-serif;
  z-index: 9999;
  opacity: 0;
  transition: opacity 0.3s ease-in-out;
}

.mensaje-temporal.mostrar {
  opacity: 1;
}
```

```
.calorias-container {  
  display: flex;  
  justify-content: space-around;  
  align-items: center;  
  margin-top: 20px;  
}  
  
.calorias-card {  
  background-color: #f5f5f5;  
  border-radius: 8px;  
  padding: 20px;  
  text-align: center;  
  box-shadow: 0 2px 4px rgba(18, 11, 11, 0.1);  
  margin: 0 10px;  
}  
  
.calorias-icon {  
  font-size: 40px;  
  color: #ff6347;  
}  
  
.calorias-value {  
  font-size: 24px;  
  font-weight: bold;  
  margin-top: 10px;  
}  
  
#map {  
  height: 90%;  
  width: 100%;  
}
```

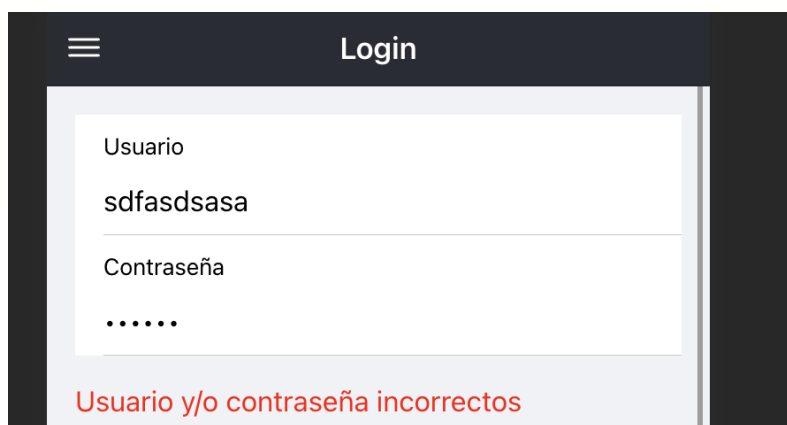
CAPTURAS APP:

Login:



A screenshot of a mobile application's login screen. The screen has a dark blue header with a hamburger menu icon on the left and the word "Login" in the center. Below the header is a light gray box containing two input fields: "Usuario" and "Contraseña". Below these fields is a blue button with the text "Iniciar sesión".

Login con error:



A screenshot of the same mobile application's login screen, but showing an error state. The "Usuario" field contains the text "sdfasdsasa" and the "Contraseña" field contains six dots. At the bottom of the screen, there is a red error message that reads "Usuario y/o contraseña incorrectos".

Registro de usuario:

The screenshot shows a mobile application interface for user registration. At the top, there is a dark header with a hamburger menu icon on the left and the title 'Registro en la aplicación' in the center. Below the header is a light gray container with a white background. Inside this container, there are four input fields: 'Usuario', 'Contraseña', 'País de residencia' (which includes a dropdown arrow icon), and 'Calorías diarias'. At the bottom of the container is a prominent blue button with the text 'Registrar Usuario' in white.

Registro existente:

This is a partial screenshot of the same registration form. It shows the top portion of the interface, including the header and the 'Usuario' and 'Contraseña' input fields. The 'País de residencia' field and the 'Registrar Usuario' button are not visible in this cropped view.

Registro exitoso:

Registro en la aplicación

Usuario

Contraseña

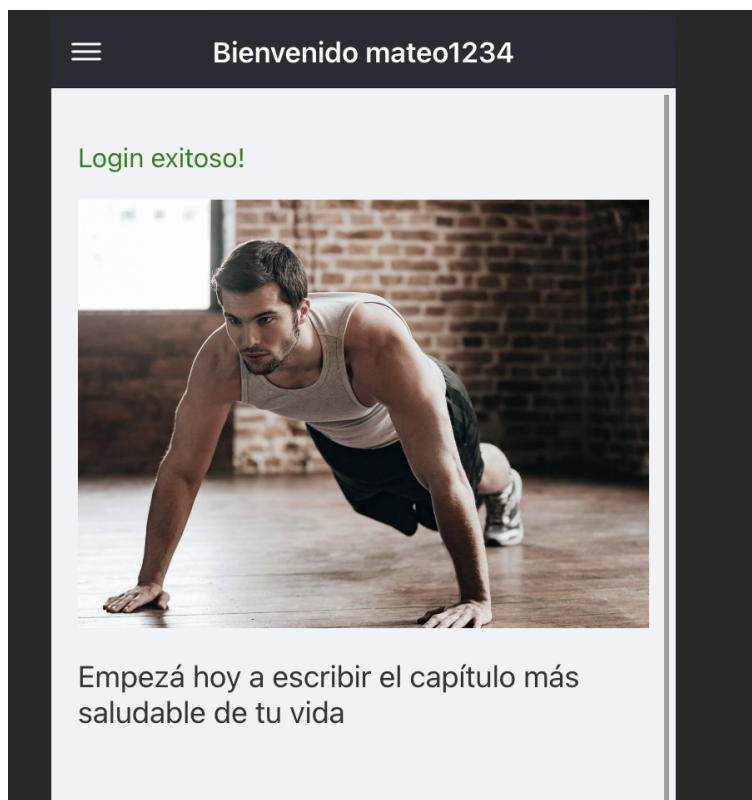
País de residencia

Calorías diarias

Registrar Usuario

Registro exitoso!

Login exitoso + Pantalla Home:



Registro de comida:

The screenshot shows a mobile application interface for registering food. The title bar is dark with a hamburger menu icon and the text 'Registrar Comida'. The form has three input fields: 'Seleccione alimento' with a dropdown arrow, 'Cantidad' with a numeric input, and a date field showing '19/02/2024' with a calendar icon. A blue button labeled 'Registrar Alimento' is at the bottom.

Registrar Comida

Seleccione alimento

Manzana

Cantidad

1

19/02/2024

Registrar Alimento

This screenshot shows the same 'Registrar Comida' form, but with a date picker overlay. The date field now shows 'dd/mm/aaaa'. The date picker is for February 2024 and has the 20th highlighted. The 'Registrar Alimento' button is partially visible behind the date picker.

Registrar Comida

Seleccione alimento

Cantidad


dd/mm/aaaa








febrero de 2024

L	M	M	J	V	S	D
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	1	2	3
4	5	6	7	8	9	10


Borrar Hoy

Listado de comida:


 **Listado de Comidas**

	Zanahoria - 41 calorías	Eliminar
	Broccoli - 55 calorías	Eliminar
	Broccoli - 55 calorías	Eliminar
	Manzana - 52 calorías	Eliminar
	Pechuga de Pollo - 165 calorías	Eliminar
	Arroz Integral - 218 calorías	Eliminar
	Manzana - 52 calorías	Eliminar

Desde:

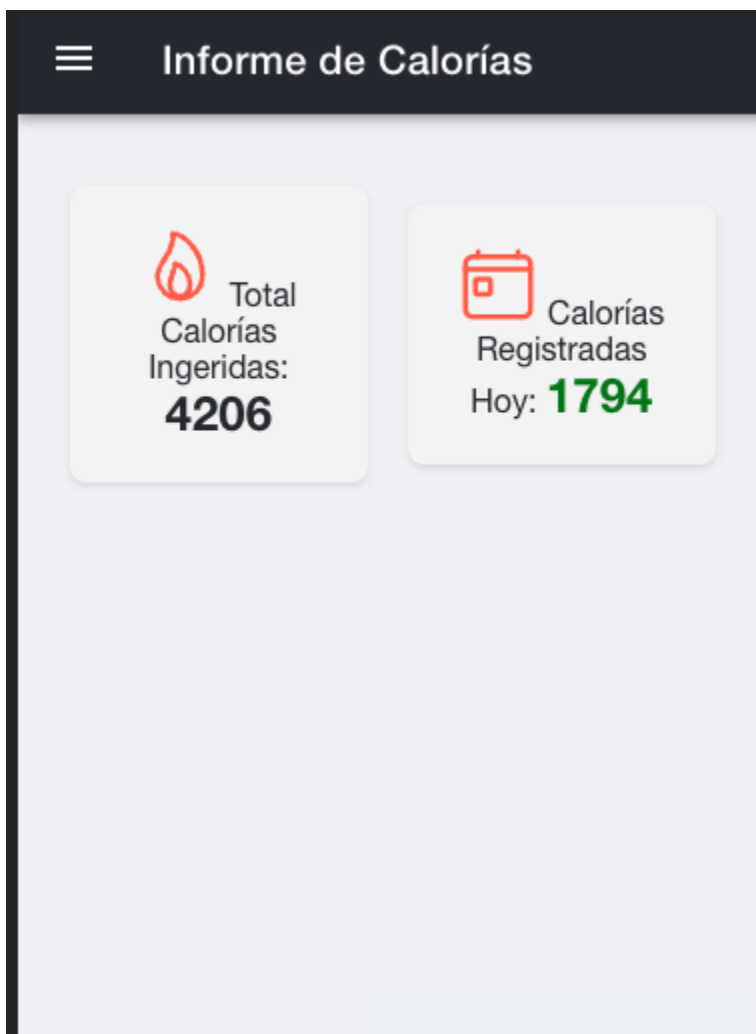
dd/mm/aaaa 

Hasta:

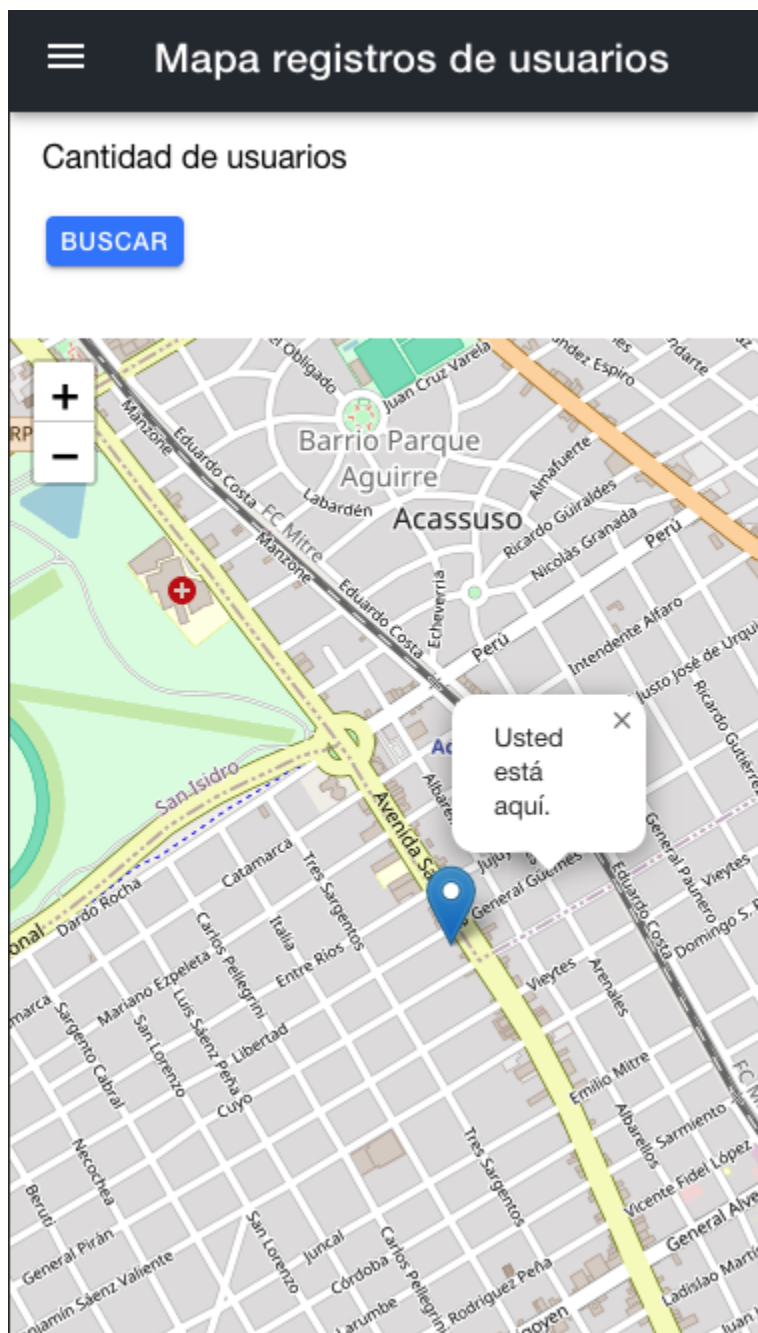
dd/mm/aaaa 

[Filtrar](#)

Informe de calorías:



Mapa ubicación usuario:



Mapa con ubicación con más registros ingresados:

