

PROYECTO COVID19

EQUIPO 1

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                }}
              </ProductConsumer>
            </div>
          </div>
        </div>
      </React.Fragment>
```

INDICE DE CONTENIDOS

1. NUESTRO CLIENTE	3
2. TECNOLOGIAS OCUPADAS	6
3. ESTRUCTURA DEL PROYECTO	7
4. DESARROLLO HITO I	8
5. RESULTADO FINAL HITO I	17
6. DESARROLLO HITO II	19
7. RESULTADO FINAL HITO II	27

PROYECTO COVID19

NUESTRO CLIENTE

CLIENTE

Departamento de estadísticas de la Organización Mundial De La Salud.

Nos contactó para ayudarlos en su estrategia del combate al Covid 19.

Encomendándonos el desarrollo de una plataforma, en la que se pudiesen visualizar los índices actualizados de la pandemia en el Mundo.

Esta plataforma debe entregar los últimos datos obtenidos desde la OMS y desplegar estos datos de forma sencilla, dinámica y fácil de entender para que pueda ser presentado a la prensa.

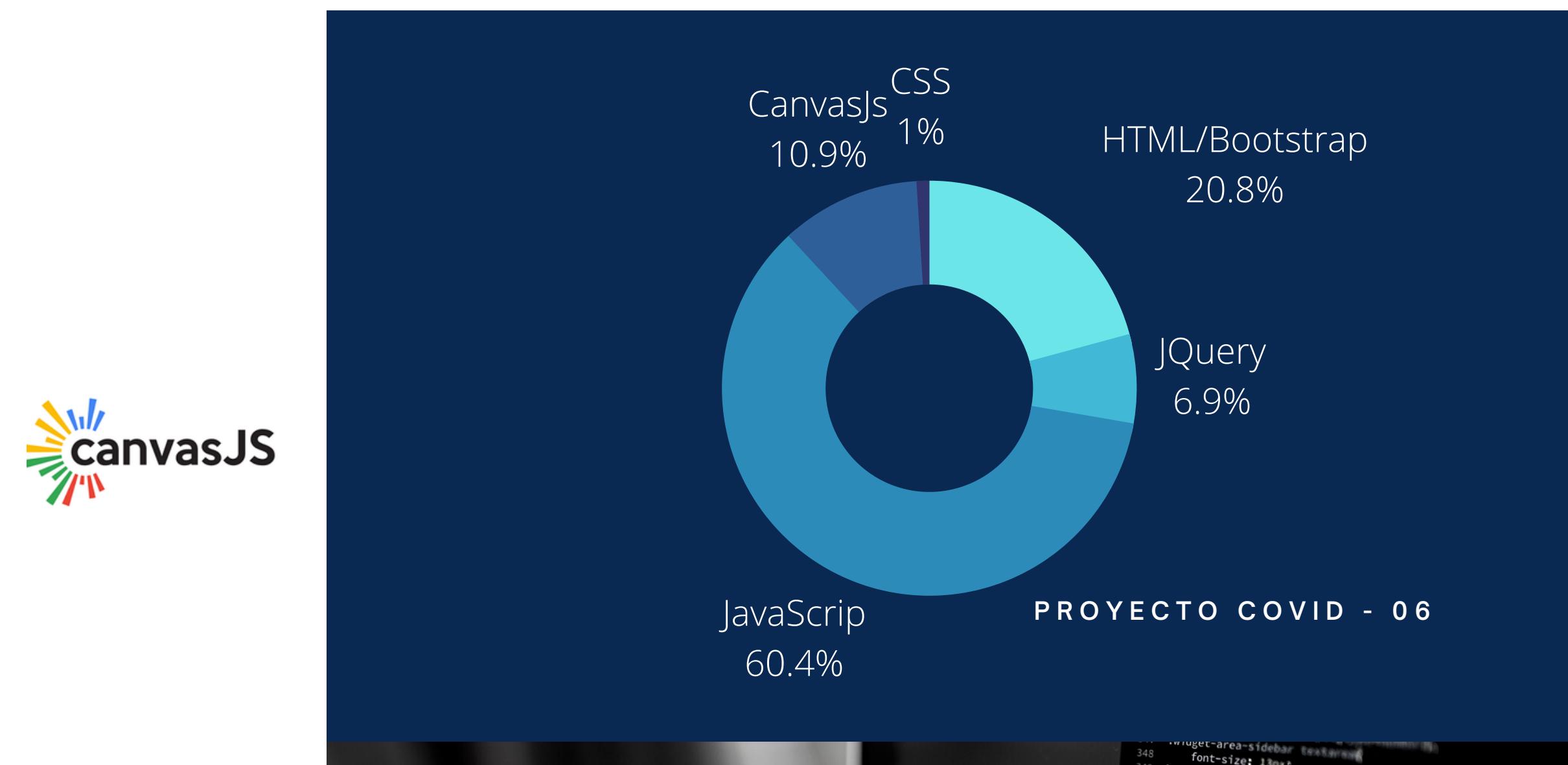
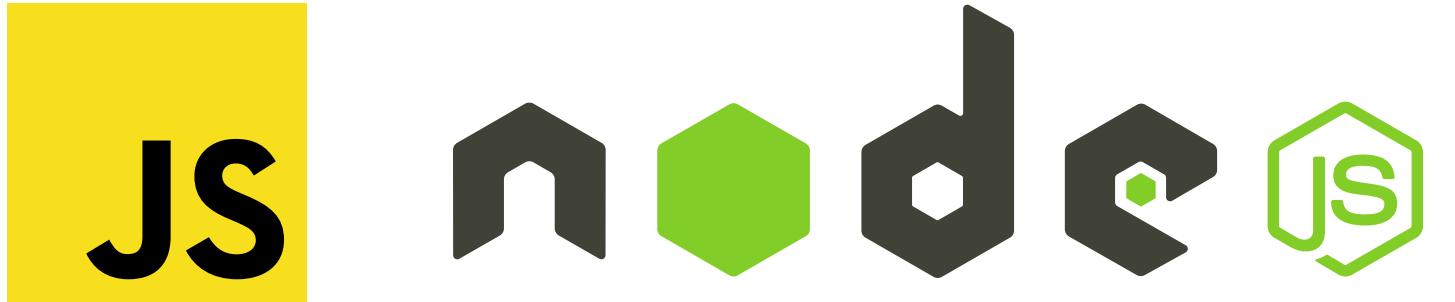
Tras varias reuniones hemos concebido el siguiente proyecto.



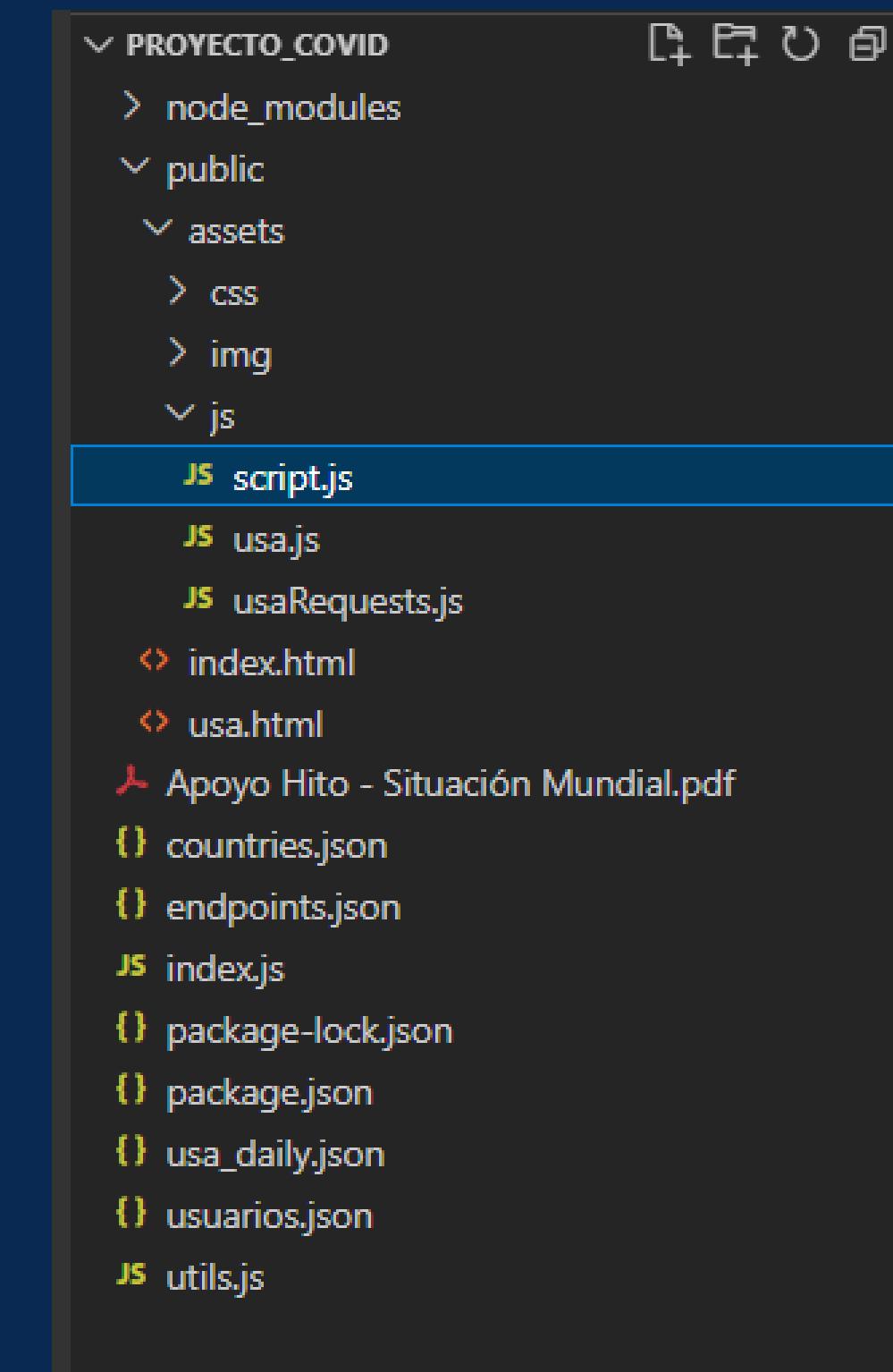
DESARROLLO

{ }

DESARROLLADO CON:



ESTRUCTURA DEL PROYECTO



The screenshot shows a file explorer window with the following structure:

- PROYECTO_COVID
 - node_modules
 - public
 - assets
 - css
 - img
 - js
 - script.js
 - usa.js
 - usaRequests.js
 - index.html
 - usa.html
 - Apoyo Hito - Situación Mundial.pdf
 - countries.json
 - endpoints.json
 - index.js
 - package-lock.json
 - package.json
 - usa_daily.json
 - usuarios.json
 - utils.js

HITO 1

SITUACIÓN MUNDIAL



```
let totalPaises = "";

let datosApi = () => {
  const urlBase = "http://localhost:3000/api/total";
  try {
    const getData = async () => {
      const response = await fetch(urlBase);
      const data = await response.json();
      totalPaises = JSON.parse(JSON.stringify(data))
      return data;
    };
    return {getData};
  } catch (error) {
    console.log(error);
  }
}();
```

SCRIPT.JS

LLAMADO A LA API

Realizamos un llamado a la API Total, que almacena todos los datos referentes a los casos COVID a nivel mundial.

Implementamos Try & Catch para poder filtrar cualquier error que se pueda presentar durante la consulta del getData.

```
const graficoTotal = (async () => [

    const data = await datosApi.getData();
    let datos = JSON.parse(JSON.stringify(data));
    totalPaises = data;
    datos.sort((a, b) => b.active - a.active);
    let recorte = datos.slice(0, 10);
    //return datos;

    let fallecidos = [];
    for (let i in recorte) {
        fallecidos.push({label: recorte[i].country, y: recorte[i].deaths})
    }

    let recuperados = [];
    for (let i in recorte) {
        recuperados.push({label: recorte[i].country, y: recorte[i].recovered})
    }

    let confirmados = [];
    for (let i in recorte) {
        confirmados.push({label: recorte[i].country, y: recorte[i].confirmed})
    }

    let activos = [];
    for (let i in recorte) {
        activos.push({label: recorte[i].country, y: recorte[i].active})
    }
]
```

FUNCION ASÍNCRONA

La función "graficoTotal" obtiene los 10 países con más casos activos del mundo y los ordena alfabéticamente con la función sort().

```
const grafico = () => {
    var chart = new CanvasJS.Chart("chartContainer", {
        animationEnabled: true,
        title: {
            text: "Situacion mundial Covid-19"
        },
        axisY: {
            title: "Casos",
            includeZero: true
        },
        legend: {
            cursor: "pointer",
            itemclick: toggleDataSeries
        },
        toolTip: {
            shared: true,
            content: toolTipFormatter
        },
        data: [
            {
                type: "bar",
                showInLegend: true,
                name: "Fallecidos",
                color: "#CD6155",
                dataPoints: fallecidos
            },
            {
                type: "bar",
                showInLegend: true,
                name: "Recuperados",
                color: "#1ABC9C",
                dataPoints: recuperados
            },
            {
                type: "bar",
                showInLegend: true,
                name: "Activos",
                color: "#F39C12",
                dataPoints: activos
            },
            {
                type: "bar",
                showInLegend: true,
                name: "Confirmados",
                color: "#3498DB",
                dataPoints: confirmados
            }
        ]
    });
    chart.render();
}
```

GRÁFICO DE BARRAS

En la función "grafico" insertamos los datos previamente filtrados a un gráfico implementando CanvasJS.

Separando por cantidad de: Fallecidos, Recuperados, Activos y Confirmados.

```

function toolTipFormatter(e) {
    var str = "";
    var total = 0;
    var str3;
    var str2;
    for (var i = 0; i < e.entries.length; i++) {
        var str1 = "<span style= \"color:" + e
            .entries[i]
            .dataSeries
            .color + "\">" + e
            .entries[i]
            .dataSeries
            .name + "</span>: <strong>" + e
            .entries[i]
            .dataPoint
            .y + "</strong> <br/>";
        total = e
            .entries[i]
            .dataPoint
            .y + total;
        str = str.concat(str1);
    }
    str2 = "<strong>" + e
        .entries[0]
        .dataPoint
        .label + "</strong> <br/>";
    str3 = "<span style = \"color:Tomato\">Total: </span><strong>" + total + "</str" +
        "ong><br/>";
    return (str2.concat(str)).concat(str3);
}

function toggleDataSeries(e) {
    if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible) {
        e.dataSeries.visible = false;
    } else {
        e.dataSeries.visible = true;
    }
    chart.render();
}
mostrarTabla();
grafico();
})();

```

El gráfico también cuenta con algunas funciones propias adicionales.

- Cuadro con cifras totales por país
- Ocultar o mostrar renglones.

```
const mostrarTabla = (async () => {
  const data = await datosApi.getData();
  let datos = JSON.parse(JSON.stringify(data));
  let paises = datos.sort(function (a, b) {
    let x = a
      .country
      .toLowerCase();
    let y = b
      .country
      .toLowerCase();
    if (x < y) {
      return -1;
    }
    if (x > y) {
      return 1;
    }
    return 0;
  });
  let tablaDatos = document.getElementById("tablaDatos");

  let tablaHead = `<thead>
    <tr>
      <th>Locación</th>
      <th>Confirmados</th>
      <th>Fallecidos</th>
      <th>Recuperados</th>
      <th>Activos</th>
      <th>Detalles</th>
    </tr>
  </thead>`;

  for (let i = 0; i < paises.length; i++) {
    tablaHead += `<tr>
      <td>${paises[i]
      .country}</td>
      <td>${paises[i]
      .confirmed}</td>
      <td>${paises[i]
      .deaths}</td>
      <td>${paises[i]
      .recovered}</td>
      <td>${paises[i]
      .active}</td>
      <td><button onclick="mostrarModal('${paises[i]
      .country}')" id='btnMostrarModal${i}' type="button" class="btn btn-primary btn-sm">Ver Detalle</button></td>
    </tr>
  `;
  }

  tablaDatos.innerHTML = tablaHead;
})
```

TABLA DE DATOS

Esta función se encarga de preparar los datos y organizarlos para finalmente presentarlos en una tabla que contiene la información detallada por país referente a la situación COVID a nivel mundial.

```
const mostrarModal = (pais) => {
    console.log(mostrarModal);
    let paisEncontrado = totalPaises.find(country => country.country == pais);
    $('#modalDetalles').modal("show");

    CanvasJS.addCultureInfo("es",
        {
            decimalSeparator: ",",
            digitGroupSeparator: ".",
        });
}

var chart = new CanvasJS.Chart("graficoModal", {
    animationEnabled: true,
    title: {
        text: "Situacion Pais"
    },
    culture: "es",
    data: [
        {
            type: "pie",
            startAngle: 240,
            //yValueFormatString: "#,###.##",
            indexLabel: "{label} {y}",
            dataPoints: [
                {
                    label: "Fallecidos",
                    y: paisEncontrado.deaths
                },
                {
                    label: "Recuperados",
                    y: paisEncontrado.recovered
                },
                {
                    label: "Activos",
                    y: paisEncontrado.active
                },
                {
                    label: "Confirmados",
                    y: paisEncontrado.confirmed
                }
            ]
        }
    );
    chart.render();
};
```

SCRIPT.JS

MODAL DETALLE POR PAÍS

Un modal con un gráfico se desplegará por cada país con su respectivo detalle.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6
7     <!-- Bootstrap css -->
8     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9g59baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>
9
10    <title>Casos Covid 19</title>
11
12  </head>
13
14  <body>
15    <!-- Barra navegacion -->
16    <nav class="navbar navbar-expand-lg navbar-dark bg-dark text-white">
17      <div class="container-fluid">
18        <h3 class="navbar-brand col-sm-8">Casos Covid</h3>
19        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
20          <span class="navbar-toggler-icon"></span>
21        </button>
22        <ul class="navbar-nav ml-auto">
23          <li class="nav-item mx-4">
24            <a href="usa.html" class="nav-link">Casos USA</a>
25          </li>
26        </ul>
27      </div>
28    </nav>
29
30    <!-- Grafico principal -->
31    <div class="container my-3 py-3">
32      <div id="chartContainer" style="height: 500px; width: 100%"></div>
33    </div>
34
35    <!-- Tabla general paises -->
36    <div class="container text-center mt-5 mb-3">
37      <h2>Detalle por país</h2>
38      <div class="row">
39        <div class="col-12">
40          <table class="table table-bordered m-3" id="tablaDatos"></table>
41        </div>
42      </div>
43    </div>
```

INDEX.HTML

VISTA CASOS COVID-19

Esta vista contiene la estructura principal del proyecto.

La barra de navegación, el gráfico que despliega los datos de los países en el top 10 y la tabla que contiene a todos los países y que además permite desplegar la información de cada uno de ellos a través de un gráfico.

```
44      <!-- Modal detalles -->
45      <div id="modalDetalles" class="modal" tabindex="-1">
46          <div class="modal-dialog">
47              <div class="modal-content">
48                  <div class="modal-header">
49                      <h5 class="modal-title" id="exampleModalLabel">...</h5>
50                      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
51                          <span aria-hidden="true">&times;</span>
52                      </button>
53                  </div>
54                  <div class="modal-body">
55                      <div id="graficoModal" style="height: 400px"></div>
56                  </div>
57                  <div class="modal-footer">
58                      <button type="button" class="btn btn-secondary" data-dismiss="modal">
59                          Cerrar
60                      </button>
61                  </div>
62              </div>
63          </div>
64      </div>
65
66      <!-- CSS Modal -->
67      <style>.modal .canvasjs-chart-container canvas {width: 100% !important;}.canvasjs-chart-credit {display: none;}</style>
68
69
70      <!-- JQuery - Bootstrap 4.5.3 -->
71      <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSS5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCxRkfv" crossorigin="anonymous"></script>
72      <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe" crossorigin="anonymous"></script>
73
74      <!-- Canvas js -->
75      <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
76
77      <!-- JS Local -->
78      <script src=".//assets/js/script.js"></script>
79
80  </body>
81 </html>
```

INDEX.HTML

Se definen e instalan clases y estilos con Bootstrap que son implementadas con sus respectivas librerías en este archivo HTML.

Se implementa también CanvasJS que permite la inclusión y funcionamiento de los correspondientes gráficos, del mismo modo se realiza el enlace con el código correspondiente

A las etiquetas principales se les asigna un ID que posteriormente nos permite su manipulación a través del DOM como también de JQuery que igualmente se encuentra implementado en este archivo.

RESULTADO FINAL

HITO 1

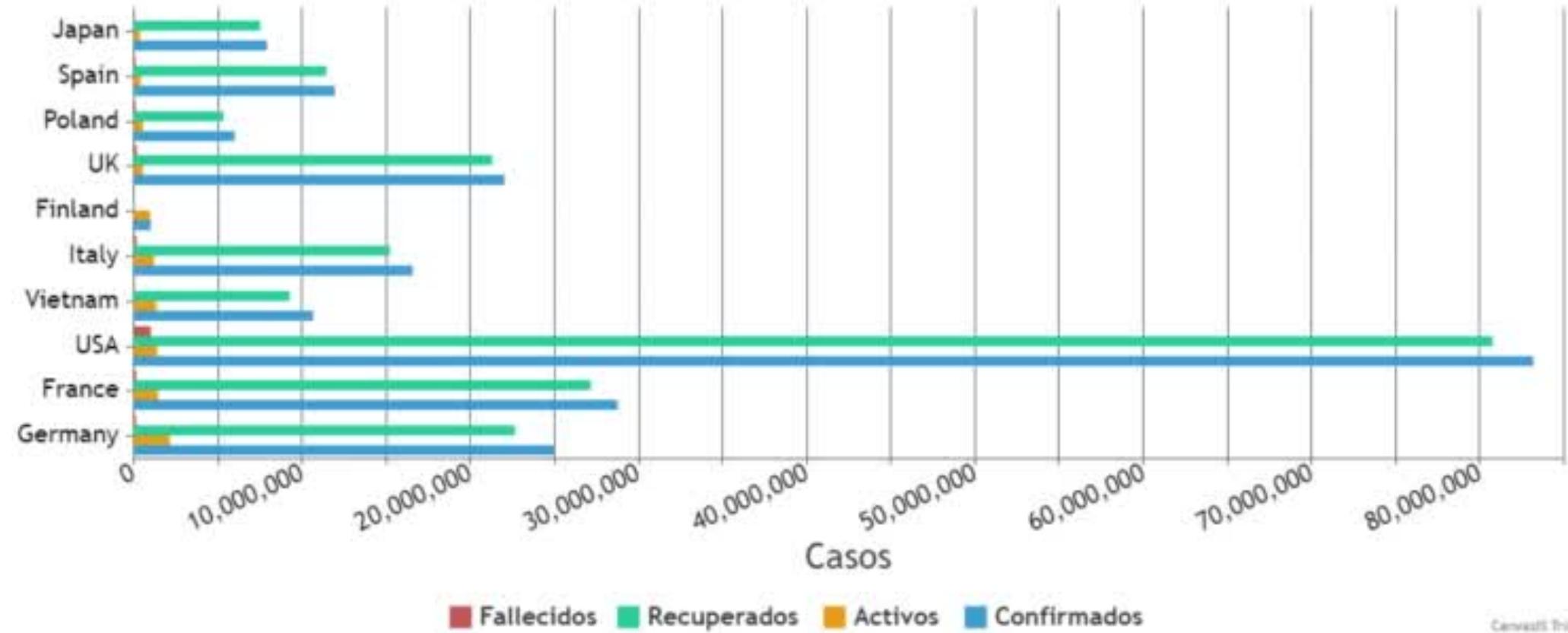
Francis | Día 2 | Día 2 | Día 4 | Día 4 | Apoyo | Lectura | Prueba | INFOC | Prezi | Inicio | Diseño | Diseño | Mis Cu | Recibido | Casos | + | - | X

127.0.0.1:5500/public/index.html

Student Dashboard Aprender: panel de... GitHub Laboral Sitios Laboral JavaScript Ejercicios Python Infoclub Python Enero Codingdojo Python Tecnologías Otros marcadores

Casos Covid Casos USA

Situación mundial Covid-19



Detalle por país

Locación	Confirmados	Fallecidos	Recuperados	Activos	Detalles
Afghanistan	178901	7683	161896	9322	<button>Ver Detalle</button>
Albania	275177	3496	271306	375	<button>Ver Detalle</button>
Algeria	265782	6875	178358	80549	<button>Ver Detalle</button>

HITO 2 SITUACIÓN USA



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

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Casos Covid 19</title>

    <!-- Bootstrap css -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
        integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous" />
</head>

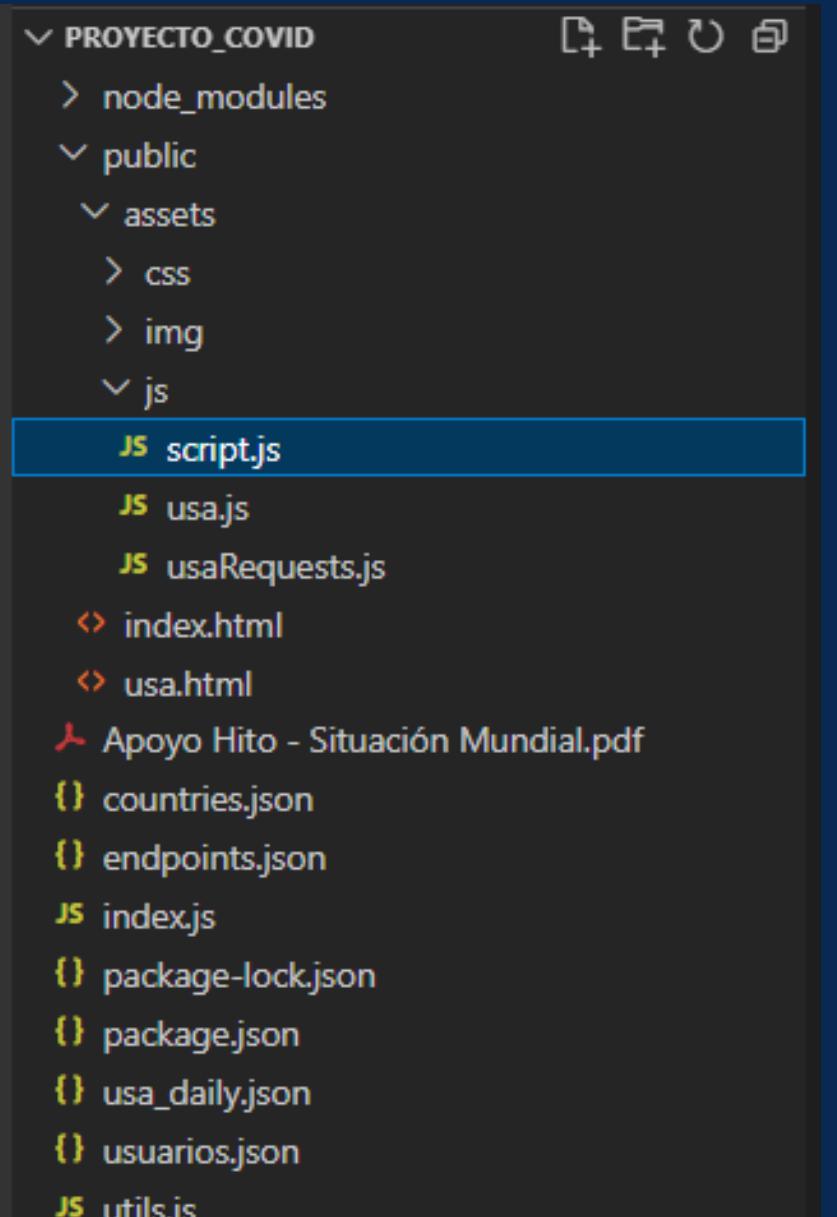
<body>
    <!-- Barra navegacion -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark text-white">
        <div class="container-fluid">
            <h3 class="navbar-brand col-sm-8">Casos Covid | USA</h3>
            <ul class="navbar-nav ml-auto">
                <li class="nav-item mx-4">
                    <a href="index.html" class="nav-link">Home</a>
                </li>
                <li class="nav-item">
                    <a id="cerrarSesion" class="nav-link" style="display: none;">Cerrar sesion</a>
                </li>
            </ul>
        </div>
    </nav>
```

CASOS COVID | USA

En el caso de esta vista, se implementará un formulario que requerirá un usuario y una contraseña para desplegar la información detallada de Estados Unidos.

USA.HTML

ESTRUCTURA DEL PROYECTO



```
<!-- formulario -->
<div class="container mt-5" style="width: 50%" id="formContainer">
  <form id="formulario">
    <div class="mb-3 form-group">
      <label for="email" class="form-label">Email</label>
      <input type="email" class="form-control" id="email" aria-describedby="emailHelp">
    </div>
    <div class="mb-3 form-group">
      <label for="password" class="form-label">Contraseña</label>
      <input type="password" class="form-control" id="password">
    </div>
    <button type="submit" class="btn btn-primary btn-sm" id="btnEnviar">Enviar</button>
  </form>
</div>

<!-- Grafico -->
<div class="container my-3 py-3">
  <div id="chartContainer" style="height: 500px; width: 100%;"></div>
</div>

<!-- JQuery - Bootstrap 4.5.3 -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
       integrity="sha384-DfXdz2htPH0lSSss5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
       crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
       integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeDyx"
       crossorigin="anonymous">
</script>

<!-- Canvas js -->
<script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>

<!-- JS Local -->
<script type="module" src="./assets/js/usa.js"></script>
</body>
</html>
```

USA.HTML

```
> assets > js > JS usaRequests.js > ...
// acceso al login y solicitud de token
const login = (email, password) => {
  return new Promise((resolve, reject) => {
    let myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");

    let raw = JSON.stringify({email: email, password: password});

    var requestOptions = {
      method: 'POST',
      headers: myHeaders,
      body: raw
    };

    fetch("http://localhost:3000/api/login", requestOptions)
      .then(
        (response) => response.text()
      )
      .then(result => {
        localStorage.setItem("Token", result);
        resolve(result);
      })
      .catch((error) => {
        reject({error, message: "Error al consultar API/LOGIN"})
      })
  })
}

let token = localStorage.getItem("Token");
```

LOGIN Y TOKEN

Implementamos una promesa, para acceder a la API Login, la cual nos permitira acceder a los datos de los usuarios registrados en la base de datos, y nos entregara un token para validarlos.

USAREQUESTS.JS

```

// formatea la fecha para regresala separada por guiones
const formatearFecha = (fecha) => {
  let fechaOriginal = fecha.toString();
  let fechaFormateada = [
    fechaOriginal.slice(0, 4),
    "-",
    fechaOriginal.slice(4, 6),
    "-",
    fechaOriginal.slice(6, 8)
  ]
  .join("");
  return fechaFormateada;
};

// ingresando y solicitando informacion de usa
const getData = (token) => {
  return new Promise((resolve, reject) => {
    let myHeaders = new Headers();
    myHeaders.append("Authorization", "Bearer " + token);

    let requestOptions = {
      method: "GET",
      headers: myHeaders
    };

    fetch("http://localhost:3000/api/country/usa", requestOptions)
      .then(
        (response) => response.json()
      )
      .then((result) => {
        let datos = result.map((registro) => {
          registro.date = formatearFecha(registro.date);
          return registro;
        });
        resolve(datos);
      })
      .catch((error) => {
        reject({error, message: "Error al consultar API/USA"})
      })
  });
};

export { login, getData, token };

```

USAREQUESTS.JS

OBTENIENDO INFORMACION USA

Mediante el uso de una promesa a la cual se le asigna el token obtenido, ingresamos y solicitamos la información de la API USA.

Tambien integramos una función que nos permite añadir guiones a la fecha por defecto que nos entrego la API.

```
import {login, getData, token} from "./usaRequests.js";

formulario.addEventListener("submit", (evento) => {
    evento.preventDefault();

    let email = document
        .getElementById("email")
        .value;
    let password = document
        .getElementById("password")
        .value;

    login(email, password).then((token) => {
        getData(token).then((data) => {
            })
        });
    });

    habilitaBtn();
    grafico();
});

const habilitaBtn = () => {
    if (token) {
        $("#formContainer").hide();
        $("#cerrarSesion").show();
        $("#chartContainer").show();
    } else {
        alert("Nombre de usuario o contraseña inválidos, por favor intente nuevamente");
    }
};
```

FORMULARIO

Se realiza un formulario de ingreso que solicitará un email y una contraseña, para poder visualizar los datos estadísticos de la situación COVID del país USA

USA.JS

```
// Grafico Situacion USA
const grafico = async () => {
    const data = await getData(token);

    let positive = [];
    let negative = [];
    let deaths = [];

    data.forEach(e => {
        positive.push({x:new Date(e.date), y:e.positive});
        negative.push({x:new Date(e.date), y:e.negative});
        deaths.push({x:new Date(e.date), y:e.deaths});
    })

    let chart = new CanvasJS.Chart("chartContainer", {
        animationEnabled: true,
        title: {
            text: "Covid-19 estadisticas USA"
        },
        legend: {
            cursor:"pointer",
            itemclick : toggleDataSeries
        },
        data: [
            {
                type: "line",
                name: "Confirmados",
                showInLegend: true,
                dataPoints: positive
            }, {
                type: "line",
                name: "Negativo",
                showInLegend: true,
                dataPoints: negative
            }, {
                type: "line",
                name: "Fallecidos",
                showInLegend: true,
                dataPoints: deaths
            }
        ]
    });
    chart.render();

    function toggleDataSeries(e) {
        if (typeof(e.dataSeries.visible) === "undefined" || e.dataSeries.visible ){
            e.dataSeries.visible = false;
        } else {
            e.dataSeries.visible = true;
        }
        chart.render();
    }
}
```

U S A . J S

G R A F I C O U S A

Se asigna un gráfico que obtendrá la información de la API, con los datos de:

Numero de Confirmados, Negativos y Fallecidos del pais.

```
$("#cerrarSesion").on("click", (e) => {
    e.preventDefault();
    localStorage.clear();
    window
        .location
        .reload();
    $("#cerrarSesion").hide();
    $("#formContainer").show();

    document
        .getElementById("email")
        .value = "";
    document
        .getElementById("password")
        .value = "";
})
```

USA.JS

CIERRE DE SESION

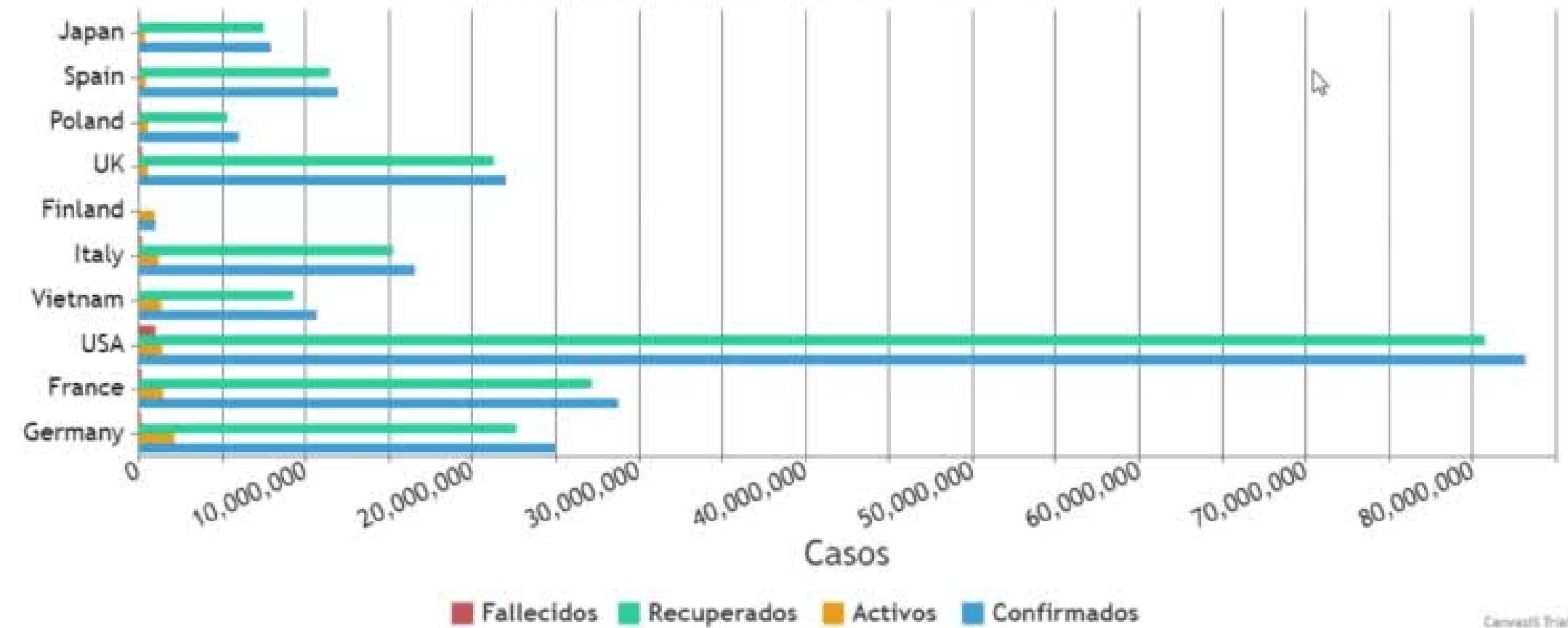
Mediante el uso de un método JQuery se logra el cierre de sesión al hacer click en el botón que se presenta en la barra de navegación una vez iniciada la sesión, y a su vez.

- Vacía el almacenamiento local (token)
- Oculta el botón cerrar sesión
- Muestra el formulario de inicio de sesión
- Limpia los campos email y contraseña

RESULTADO FINAL

HITO 2

Situacion mundial Covid-19



Detalle por país

Locación	Confirmados	Fallecidos	Recuperados	Activos	Detalles
Afghanistan	178901	7683	161896	9322	Ver Detalle
Albania	275177	3496	271306	375	Ver Detalle
Algeria	265782	6875	178358	80549	Ver Detalle

FIN