**Javascript II**

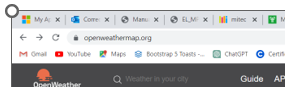
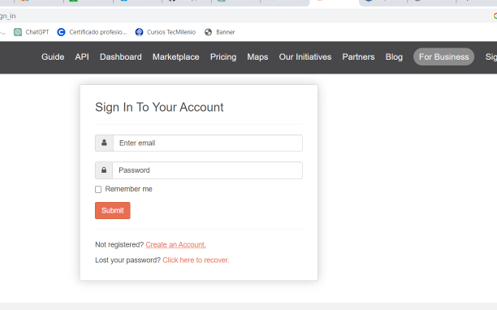
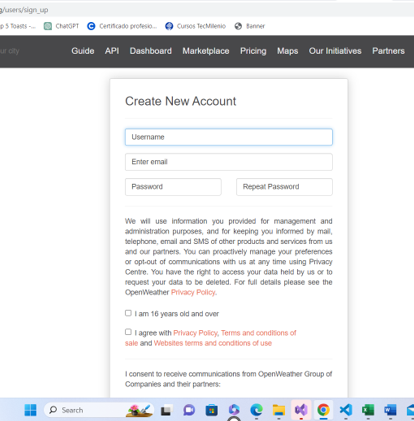
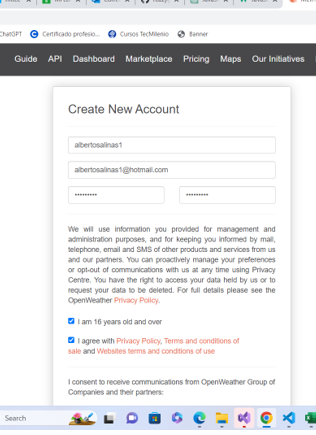
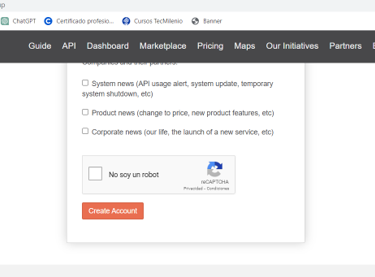
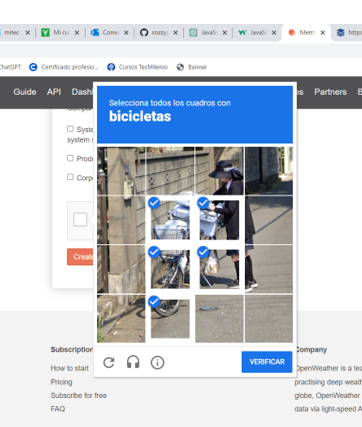
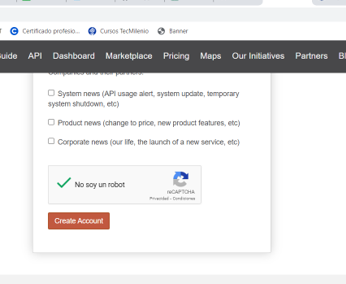
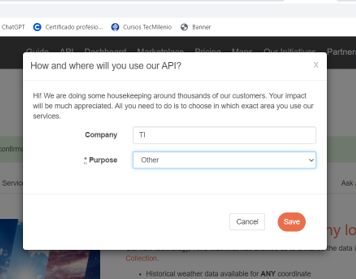
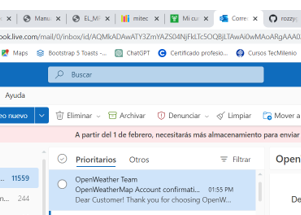
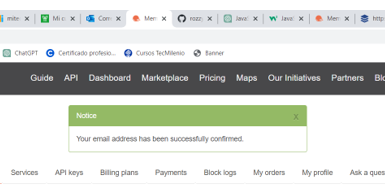
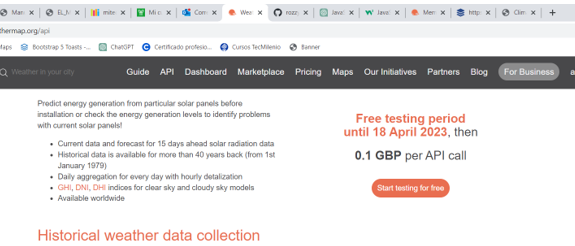
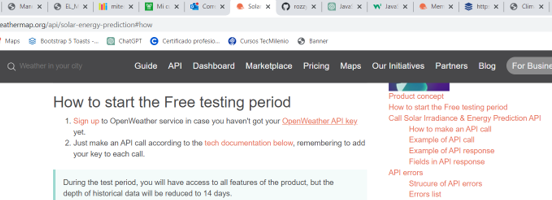
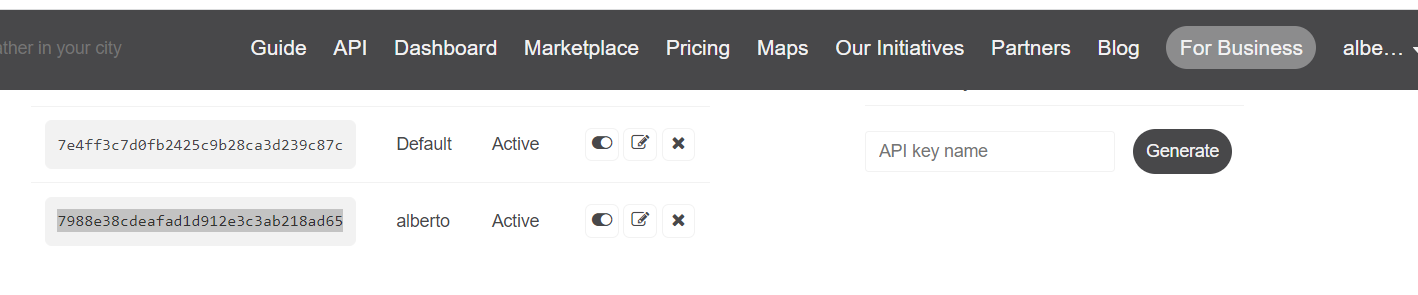
**Proyecto**

Objetivo: Crear una página simple que funcionalmente, por medio de código Javascript permita mostrar las condiciones del clima de acuerdo a una ciudad o grupo de ciudades capturada y mostrando y conservando los resultados de búsqueda en la misma pantalla.

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actividad** | **Referencia** |
| 1 | Instala las siguientes 3 extensiones en tu Visual Studio Code:   * EsLint, * LiveServer y * Prettier   (en la imagen aparece Dev Containers, favor de ignorar ese elemento).  Las 3 extensiones mencionadas nos ayudarán a mostrar mejor nuestro código y atener la opción de ejecutar y probar nuestros resultados mientras modificamos código. |  |
| 2 | Crea un folder donde guardes tu proyecto, puede ser cualquier nombre que tu decidas, por ejemplo, “Clima”.  En este folder se guardará el archivo html, css y el de nuestro código de javascript. |  |
| 3 | Crea dos archivos con el primer icono o gráfico con el signo “+”:  index.html y script.js |  |
| 4 | Empieza ubicando el archivo index.html para agregar el código base inicial, empezando por el head indicando el charset UTF-8 y un título “Consulta de Clima”. | <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>Consulta de Clima</title>  </head> |
| 5 | Dentro de la misma etiqueta head agrega un estilo básico para dar valores a las propiedades de la tabla que se utilizará para colocar los resultados de búsqueda, estas propiedades son border, padding y text-align. | <style>  table {  border-collapse: collapse;  }  th, td {  border: 1px solid black;  padding: 8px;  text-align: left;  }  </style> |
| 6 | En el body agregar una etiqueta h1 para mostrar un encabezado: “Consulta de Clima”. | <body>  <h1>Consulta de Clima</h1> |
| 7 | Crear 2 etiquetas div donde la primera contendrá un label y la segunda una caja de texto y un botón para la búsqueda de clima por nombre de ciudad especifica.  Puedes seleccionar los identificadores que gustes, se da un ejemplo en el código referencia, solo recuerda mencionarlos correctamente más adelante en el código javascript si es que los cambias.  En el evento onclick del botón manda llamar la función consultarClima () que se codificará más adelante. | <div>  <label for="ciudad">Ciudad :</label>  </div>  <div>  <input type="text" id="ciudad">  <button onclick="consultarClima()">Consultar</button>  </div> |
| 8 | Crear 2 etiquetas div donde la primera contendrá un label y la segunda una caja de texto y un botón para la búsqueda del clima, pero de varias ciudades, por nombre de ciudad especifica, separadas por coma (esta sería la sintaxis a seguir para buscar el clima de varias ciudades a la vez).  Puedes seleccionar los identificadores que gustes, se da un ejemplo en el código referencia, solo recuerda mencionarlos correctamente más adelante en el código javascript si es que los cambias.  En el evento onclick del botón manda llamar la función consultarClimas() que se codificará más adelante. (Nota que la función es muy parecida en nombre a la anterior, solo se le agregó una “s” al final para indicar que la búsqueda es en plural, es decir de varias ciudades a la vez.  En el atributo value de la caja de texto coloca 3 ciudades a desplegarse por default, precargadas para el usuario que sirvan de ejemplo. | <div>  <label for="ciudades">Ciudades:</label>  </div>  <div>  <input type="text" id="ciudades" value="London, Paris, Berlin">  <button onclick="consultarClimas()">Consultar</button>  </div> |
| 9 | Escribe 2 etiquetas br para separar 2 líneas el desplegado html. | <br><br> |
| 10 | Escribe un div y dentro coloca un botón donde en su evento onclick mandará llamar un método que realice una limpia de la tabla donde se van desplegando y acumulando los resultados de las búsquedas de climas de ciudades. El método se llamará “limpiarTabla” | <div>  <button onclick="limpiarTabla()">Limpiar</button>  </div> |
| 11 | Escribe 2 etiquetas br para separar 2 líneas el desplegado html. | <br><br> |
| 12 | Escribe un div donde se colocará una tabla que guardará y desplegará los resultados de búsqueda de clima.  Indica el id de la tabla como “tabla-clima”  Declara 3 columnas fijas con nombre “Ciudad”, “Temperatura”, “Descripción” | <div>  <table id="tabla-clima">  <thead>  <tr>  <th>Ciudad</th>  <th>Temperatura</th>  <th>Descripción</th>  </tr>  </thead>  <tbody>  </tbody>  </table>  </div> |
| 13 | Por último, como última línea antes de cerrar el body, inserta la etiqueta script que indique donde ira a buscar el código javascript con relación a las funciones mencionadas previamente en los eventos onclick de los botones html. | <script src="script.js"></script>  </body>  </html> |
| 14 | Ahora, moviéndonos al código del otro archivo que creamos al inicio llamado script.js colocaremos el código javascript para buscar el clima de una ciudad a la vez con base en su nombre tecleado. | function consultarClima() { |
| 15 | Crea una constante donde guardes el valor de la ciudad que tecleo el usuario. | const ciudad = document.getElementById('ciudad').value; |
| 16 | Agrega la llave que ocuparemos para hacer uso de un servicio web donde obtendremos los resultados de la búsqueda del clima.  La página que ofrece este servicio es : <https://home.openweathermap.org/>  Puedes usar el código de la llave que yo utilizo, aquí la dejo, o puedes crear una tuya, en el anexo dejo los pasos para que lo hagas, pero puedes utilizar esta que te dejo en la referencia de código. | const API\_KEY = '7988e38cdeafad1d912e3c3ab218ad65'; // Sustituye "tu\_api\_key" por tu propia API key de OpenWeatherMap |
| 17 | Agrega una constante que apunte al servicio web del clima incluyendo en la url la ciudad y la llave anterior viendo la estructura en la referencia. | const url = `https://api.openweathermap.org/data/2.5/weather?q=${ciudad}&appid=${API\_KEY}`; |
| 18 | Mediante el llamado asíncrono fetch solicita el servicio web anterior preparando el código por si existe algún error en el llamado. Si responde correctamente regresar la respuesta en formato json y enseguida indicarle que despliegue dicho resultado en la tabla de despliegue html previamente creada.  Insertar los valores respectivos a cada columna de la tabla de acuerdo a la respuesta json recibida que incluye las propiedades siguientes:  name, main.temp, weather[0].description las cuales respectivamente son las que hay que llenar en la tabla.  El valor de temperatura te lo regresa en grados kelvin, realiza la siguiente conversión para obtenerlos en grados Celsius:  (data.main.temp - 273.15).toFixed(1)  Proteger este llamado fetch por si existe un error con el complemento del manejo de excepciones con la instrucción catch. | fetch(url)  .then(response => {  if (response.ok) {  return response.json();  } else {  throw new Error('Error en la respuesta de la API');  }  })  .then(data => {  // Mostrar resultado en la tabla  const tabla = document.getElementById('tabla-clima').getElementsByTagName('tbody')[0];  const fila = tabla.insertRow();  fila.insertCell().innerHTML = data.name;  fila.insertCell().innerHTML = `${(data.main.temp - 273.15).toFixed(1)}°C`;  fila.insertCell().innerHTML = data.weather[0].description;  })  .catch(error => {  console.error('Error al consultar el clima', error);  });  } |
| 19 | Ahora, crea otra función javascript para buscar el clima de varias ciudades a la vez con base a las ciudades, separadas por coma, tecleadas por el usuario. | function consultarClimas() { |
| 20 | Crea una constante donde guardes el valor de las ciudades que tecleo el usuario. | const ciudades = document.getElementById('ciudades').value.split(',').map(ciudad => ciudad.trim()); |
| 21 | Agrega la llave que ocuparemos para hacer uso de un servicio web donde obtendremos los resultados de la búsqueda del clima.  La página que ofrece este servicio es : <https://home.openweathermap.org/>  Puedes usar el código de la llave que yo utilizo, aquí la dejo, o puedes crear una tuya, en el anexo dejo los pasos para que lo hagas, pero puedes utilizar esta que te dejo en la referencia de código. | const API\_KEY = '7988e38cdeafad1d912e3c3ab218ad65'; // Sustituye "tu\_api\_key" por tu propia API key de OpenWeatherMap |
| 22 | Escribe una promesa de javascript para todas las ciudades dándoles un manejo de estructura map.  Para ir obteniendo todos los resultados de las ciudades en una variable llamada “ciudad” mediante un llamado de función (función de flecha) que llame al url, con los parámetros que el servicio web requiere (se puede ver el ejemplo de la url en la referencia) incluyendo en la url la ciudad y la llave anterior.  Preparando el código por si existe algún error en el llamado. Si responde correctamente regresar la respuesta en formato json y enseguida indicarle que despliegue dicho resultado en la tabla de despliegue html previamente creada.  Insertar los valores respectivos a cada columna de la tabla de acuerdo a la respuesta json recibida que incluye las propiedades siguientes:  name, main.temp, weather[0].description las cuales respectivamente son las que hay que llenar en la tabla.  El valor de temperatura te lo regresa en grados kelvin, realiza la siguiente conversión para obtenerlos en grados Celsius:  (data.main.temp - 273.15).toFixed(1)  Proteger este llamado fetch por si existe un error con el complemento del manejo de excpeciones con la instrucción catch. | Promise.all(ciudades.map(ciudad => {  const url = `https://api.openweathermap.org/data/2.5/weather?q=${ciudad}&appid=${API\_KEY}`;  return fetch(url).then(response => {  if (response.ok) {  return response.json();  } else {  throw new Error('Error en la respuesta de la API');  }  });  }))  .then(data => {  // Mostrar resultados en la tabla  const tabla = document.getElementById('tabla-clima').getElementsByTagName('tbody')[0];  data.forEach(ciudad => {  const fila = tabla.insertRow();  fila.insertCell().innerHTML = ciudad.name;  fila.insertCell().innerHTML = `${(ciudad.main.temp - 273.15).toFixed(1)}°C`;  fila.insertCell().innerHTML = ciudad.weather[0].description;  });  })  .catch(error => {  console.error('Error al consultar el clima', error);  });  } |
| 23 | Codifica el método que limpia la tabla html donde despliega los resultados de consultas realizadas llamada “limpiarTabla”. | function limpiarTabla() {  // Mostrar resultado en la tabla  const tabla = document.getElementById('tabla-clima').getElementsByTagName('tbody');  for(let i = 0; i<tabla.length; i++)  {  tabla[i].innerHTML = "";  }  } |
| 24 | Guarda todos tus cambios y ejecuta tu página HTML, debe de poder traer la información del web service.  Nota: Si creaste tu propia llave tarda máximo un par de horas en activarse |  |
| 25 | Depura tu código y resguárdalo. |  |

**Anexo**

Creación de llave en página <https://home.openweathermap.org/>

1. Entra a la página anterior:
2. Accede a la opción de registro para crear tu cuenta: 
3. Da de alta tu cuenta: 
4. Desplaza hacia abajo y continua con tu registro: 
5. Valida que no eres un robot: 
6. Sigue y completa instrucciones: 
7. Confirma que te valide la página; 
8. Continua con lo que te pide, te doy unos datos ejemplo: 
9. Confirma que te llegue correo de confirmación: 
10. Sigue instrucciones en el correo y deberás recibir confirmación definitiva: 
11. Desliza en la página , una vez confirmado tu registro, a la opción de prueba: 
12. Da click en la parte que mención la API Key: 
13. Por último, puedes crear tu propia llave solo escribe un texto en el campo que te indica y genera tu llave: 
14. Espera dos horas y podrás hacer uso de la llave para consultas, normalmente tarda menos tiempo.
15. Este es el ejemplo básico para consulta de clima para una ciudad:

const url = `https://api.openweathermap.org/data/2.5/weather?q=${ciudad}&appid=${API\_KEY}`;

O tecleando en el URL directo del navegador, por ejemplo:

<https://api.openweathermap.org/data/2.5/weather?q=Monterrey&appid=7988e38cdeafad1d912e3c3ab218ad65>

Para el ejemplo anterior te dará un resultado en formato json similar al siguiente:

{"coord":{"lon":-100.3167,"lat":25.6667},"weather":[{"id":802,"main":"Clouds","description":"scattered clouds","icon":"03d"}],"base":"stations","main":{"temp":307.07,"feels\_like":304.68,"temp\_min":304.12,"temp\_max":308.1,"pressure":1010,"humidity":15,"sea\_level":1010,"grnd\_level":951},"visibility":10000,"wind":{"speed":3.2,"deg":319,"gust":5.63},"clouds":{"all":40},"dt":1681587680,"sys":{"type":2,"id":2004940,"country":"MX","sunrise":1681561104,"sunset":1681607029},"timezone":-21600,"id":3995465,"name":"Monterrey","cod":200}