| **INGENIERÍA DE SISTEMAS DE INFORMACIÓN**  **Desarrollo de Software - Ciclo lectivo 2023** |
| --- |

| **Características del Práctico : Front end - HTML – CSS** | |
| --- | --- |
| **Unidad** | **2** |
| Tema: | **Front End – Introducción a TypeScript** |
| **Resultados de Aprendizajes** | RA1: Desarrolla un sistema mediante una metodología y estándares a partir de la situación real de una organización modelo, empresa o emprendimiento para obtener el: análisis, diseño, programación y verificación de componentes de software.    RA2: Aplica las estrategias de algoritmia, teniendo en cuenta el paradigma de programación adecuado y las buenas prácticas de programación para una producción de artefacto de software de buena calidad y que cumplan con los requisitos establecidos |
| **Objetivo** | El objetivo de este trabajo práctico es que los estudiantes adquieran experiencia en la creación de una página HTML básica utilizando los elementos HTML y etiquetas básicas que se han visto en clase, consuman una api pública mediante fetch API de JavaScript nativo, y escriban todo el código necesario en TypeScript para luego compilarlo a su código JavaScript correspondiente. |
| **Requisitos técnicos:** | Uso de Visual Studio Code, terminal y la consola del Browser |
| **Fecha de inicio** | 20 de Septiembre de 2023 - |
| **Fecha de entrega:** | 27 de Septiembre de 2023 - Hora: 23:00 hs |
| **Modalidad de entrega :** | Actividad agendada en aula virtual |
| **Modalidad del Trabajo Práctico** | Desarrollo Individual – Entrega en aula virtual |
|  | |
| **Enunciado**:  Como se muestra en el siguiente video: <https://recordit.co/z32RvwrVCP> queremos crear una página donde vamos a consumir un endpoint (<https://fakestoreapi.com/products>) que nos da como resultado un listado de 20 productos, para lo cual mostraremos un spinner hasta terminar de dibujar la tabla resultante mostrando su contenido y ocultando el spinner una vez terminado.  **Una posible aproximación puede ser el siguiente código**:  <!doctype html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>TS demo</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">  <style>  .loader {  width: 48px;  height: 48px;  border: 5px solid #FFF;  border-bottom-color: #FF3D00;  border-radius: 50%;  display: inline-block;  box-sizing: border-box;  animation: rotation 1s linear infinite;  }  @keyframes rotation {  0% {  transform: rotate(0deg);  }  100% {  transform: rotate(360deg);  }  }  #spinnerContainer {  align-items: center;  display: flex;  height: 100%;  justify-content: center;  position: absolute;  width: 100%;  }  #tableContainer {  padding: 20px;  }  </style>  </head>  <body>  <div id="spinnerContainer">  <span class="loader"></span>  </div>  <div id="tableContainer">  <table class="table table-bordered" id="tableElement"></table>  </div>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"  integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"  crossorigin="anonymous"></script>  <script src="./index.js"></script>  </body>  </html>  type Rating = {  count: number;  rate: number;  };  type Product = {  category: string;  description: string;  id: number;  image: string;  price: number;  rating: Rating;  title: string;  };  fetch('https://fakestoreapi.com/products')  .then(res => res.json())  .then((products: Product[]) => {  // Prepare table HTML  let tableHTML: string = '<thead><tr><th>ID</th><th>Title</th><th>Description</th><th>Price</th></tr></thead><tbody>';  // Loop thru all products to generate rows of the table  products.forEach((p: Product) => {  tableHTML += `<tr><td>${p.id}</td><td>${p.title}</td><td>${p.description}</td><td>${p.price}</td></tr>`;  });  // Close table body  tableHTML += '</tbody>';  // Grab table element to set its inner HTML  document.querySelector('#tableElement')!.innerHTML = tableHTML;  // Hide spinner  const spinnerElement: HTMLElement = document.querySelector('#spinnerContainer')!;  spinnerElement.style.display = 'none';  }); | |
| **Recursos de Boostrap:**  **Template para archivo index.html**: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>  <!doctype html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title>Bootstrap demo</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">  </head>  <body>  <h1>Hello, world!</h1>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>  </body>  </html> | |
| **Desarrollo:**  En esta sección, los estudiantes deben proporcionar detalles sobre cómo implementaron la lógica de creación de objetos y el mapeo de relaciones para demostrar que el modelo funciona correctamente en función de su descripción. A continuación se detallan los pasos y las actividades que deben llevar a cabo:   1. Crear un archivo HTML llamado "index.html" utilizando un editor de texto o una herramienta de desarrollo web. 2. En el archivo HTML, crear un contenedor para nuestro spinner y un contenedor para nuestra tabla de productos. 3. Crear un archivo index.ts donde haremos la llamada a nuestro endpoint de productos y una vez que tenemos el resultado, dibujaremos la tabla y esconderemos el spinner.   Guardar el archivo "index.html". | |
| **Conclusiones:**  La evaluación se basará en la presencia de los elementos HTML requeridos, la correcta estructura del archivo HTML, y el uso adecuado de los atributos, el uso de fetch API, escribir código en TypeScript para luego pasarlo a su código JavaScript correspondiente. No se evaluará el diseño o la apariencia de la página en este práctico,  solo la disposición de los elementos y los atributos correspondientes. | |
| Forma de Presentación | El proyecto se presentar con un link de un repositorio de proyecto en github, con todas las dependencias funcionando.  Se debe crear en el repo un archivo readme con las características de ejecución del mismo.  El link del repo se subirá al aula virtual. |
| Tipo de Evaluación | Formativa - Individual |
| **Modalidad de Evaluación** | |  |  | **>70%** | **<50 %** | | --- | --- | --- | --- | | **Estructura Html 5** |  |  |  | | **CSS Spinner** |  |  |  | | **Table Container** |  |  |  | | **Llamada fetch** |  |  |  | | **Codigo en TS** |  |  |  | | **Llenado de tabla** |  |  |  | | **Ocultar spinner** |  |  |  | | **RESULTADOS** | **Excelente** | **Satisfactorio** | **No Satisfactorio** |   **Se establece una sumatoria de cada ítem y se determina el porcentaje individual de cada uno. Posteriormente se establece la sumatoria de todos los ítems y se lo divide por la cantidad de los mismos para determinar en cuál de los valores de la escala se encuadra.**   |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |
| Bibliografía | Listas de reproducción de videos de la cátedra y material teórico del aula virtual  [**https://youtu.be/cIeYm0a33n4**](https://youtu.be/cIeYm0a33n4)  [**https://youtu.be/Eukf-zNFGx0**](https://youtu.be/Eukf-zNFGx0)  [**https://youtu.be/oaHVWwTWcEE**](https://youtu.be/oaHVWwTWcEE)  [**https://youtu.be/2G2KkLGp13s**](https://youtu.be/2G2KkLGp13s)  [**https://youtu.be/TA1Ytwk1N9I**](https://youtu.be/TA1Ytwk1N9I)  [**https://youtu.be/S66L4ZaQvgQ**](https://youtu.be/S66L4ZaQvgQ) |
| Profesores responsables del TP. | | Apellido y nombre | | Apellido y nombre | | | --- | --- | --- | --- | | Cortez,  Alberto | | Cassani  Agustin | | |  |  | |  | |  |  | |