| **UNIVERSIDAD TECNOLÓGICA FRM**  **Desarrollo de Software - Ciclo lectivo 204** |
| --- |

| **Práctico : Front End - typescript** | |
| --- | --- |
| **Unidad** | **3** |
| Tema: | **Tecnologías de Front End** |
| **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** |  |
| **Requisitos técnicos:** | Uso de Visual Studio Code y la consola del Browser |
| **Fecha de inicio** | 17 de Agosto de 2024 - |
| **Fecha de entrega:** | Actividad agendada en aula virtual |
| **Modalidad de entrega :** | Actividad agendada en aula virtual |
| **Comisiones** | **Comisiones: AMBAS COMISIONES** |
| **Modalidad del Trabajo Práctico** | Desarrollo Individual – Entrega en aula virtual |
| **Enunciado**: Creación de un proyecto html y estilos con css | |
| **Recursos de Videos:**  **link**:<https://www.youtube.com/watch?v=T7uaEZ3ZoZE> | |
| 1. **Verifica que TypeScript esté instalado globalmente en tu sistema** -- En consola deben correr node -v y tsc -v estos comandos deben devolverles las versiones en el caso de no tener instalados los elementos recurrir a los videos 2. Crea un proyecto  * Genera un archivo html * Genera un archivo index.html * Genera un archivo index.ts * comienza un proyecto de typescript con ‘npx tsc -y’ * Agrega un console.log(“Ejercicio 2: corriendo desde typescript”) * Corre el comando tsc -w * Abre el live server desde el html * Verifica que se muestra en consola tu console.log()   – En consola    –Estructura de proyecto   3. Crea variables de diferentes tipos de datos (string, number, boolean, Date). Muestra sus valores en el HTML. Utiliza los siguientes tipos: string, number, boolean, Date –vista html:   4. Crea una función que convierta un número a una cadena de texto. Muestra el resultado en el HTML. La función debe recibir un número y devolver su representación en cadena. –vista html: 5. Crea un array de números y una función que sume todos los elementos del array. Muestra la suma en el HTML. La función debe recibir un array de números y devolver la suma de todos sus elementos. –vista html:   6. Crea un objeto que representa a un estudiante con nombre, edad y curso. Muestra al estudiante y sus propiedades en el HTML. El objeto debe tener propiedades para el nombre, edad y curso del estudiante. –vista html:   7. Define un “type” personalizado para representar una dirección con calle, ciudad y código postal. Usa este tipo para crear una dirección y muéstrala en el HTML. El tipo debe tener propiedades para calle, ciudad y código postal. –vista html:   8. Define una interfaz para un usuario que tenga nombre, correo y un método para saludar. Implementa esta interfaz en un objeto y muestra el saludo en el HTML. La interfaz debe definir propiedades para nombre, correo y un método que devuelva un saludo. –vista html: 9. Crea una clase `Persona` con propiedades para nombre y edad. Implementa un método para presentarse y muestra el resultado en el HTML. La clase debe tener un constructor para inicializar nombre y edad, y un método que devuelva una cadena con la presentación. –vista html: 10. Crea una clase genérica `Caja` que almacena un valor de tipo genérico. Implementa un método para obtener el valor y muestra los resultados en el HTML. La clase debe permitir almacenar y obtener un valor de cualquier tipo. – Requisitos:   * llamar la función pasándole el type string * llamar la función pasándole el type number * verificar que el tipo de dato pasado y el parámetro se muestran correctamente   –vista html: 11. Crea una función genérica `identidad` que devuelva el mismo valor que recibe. Usa esta función para diferentes tipos y muestra los resultados en el HTML. La función debe aceptar un valor de cualquier tipo y devolverlo sin modificarlo. –vista html:   12. Define una enumeración `Color` con valores para diferentes colores. Usa esta enumeración para asignar un color favorito a una variable y muéstralo en el HTML. La enumeración debe incluir al menos tres colores diferentes. –Requisitos:   * Genera un enum con tre colores * Asigna a una variable el enum   –vista html: | |
| Forma de Presentación | El proyecto se debe presentar en un archivo txt donde en su interior se mencione el repositorio PUBLICO DE GITHUB.  Se debe crear en el repo un archivo readme con las características de ejecución del mismo.  El link del repositorio se subirá al aula virtual en el archivo mencionado.  El archivo a subir al aula virtual debe nombrarse de la siguiente manera: nombreapellidolegajo.txt  **Ejemplo: mariaLopez3456.txt** |
| Tipo de Evaluación | Formativa - Individual |
| **Modalidad de Evaluación** | |  | **>90** | **90% y 60%** | **<60 %** | | --- | --- | --- | --- | | **presentación** |  |  |  | | **nombre de variables** |  |  |  | | **Orden del código** |  |  |  | | **comentarios en el codigo** |  |  |  | | **secciones de html** |  |  |  | | **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 divide por la cantidad de los mismos para determinar en cuál de los valores de la escala se encuadra.**   |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |
| Bibliografía | Listas de reproducción de vídeos de la cátedra y material teórico del aula virtual  **link**:<https://www.youtube.com/watch?v=T7uaEZ3ZoZE>  **documentacion**: <https://www.typescriptlang.org/> |
| Profesores responsables del TP. | | Apellido y nombre | | Apellido y nombre | | | --- | --- | --- | --- | |  | |  | | |  |  | |  | |  |  | |