

Boletín de Ejercicios - Sprint 2

Descripción del ejercicio

Para asentar los conocimientos mostrados en la parte teórica, se tendrá que proceder a resolver los siguientes ejercicios/problemas en el orden correcto. Para ello, primero pasamos a explicar los criterios que debemos seguir para la resolución de los ejercicios/problemas.

Fases de la resolución de problemas

1. **Análisis del problema:** Se debe indicar en el directorio específico de la asignatura el problema que se va a resolver de una forma adecuada, es decir, no debe contener ambigüedades, debe ser simple y autocontenido.
2. **Diseño de la propuesta de solución del problema:** Como todo aquel problema que se quiere resolver, es necesario realizar el diseño de la o las soluciones que se procederá a implementar en el siguiente paso. Para esto nos debemos ayudar de las herramientas para realizar esquemas gráficos (UML, Diagramas de flujos, etc...)
3. **Implementación del diseño propuesto:** En este punto ya se procederá a implementar todo el diseño establecido en el punto anterior.
4. **Pruebas de la resolución del problema:** Es indispensable el realizar pruebas para verificar la integridad y correcto funcionamiento de la implementación realizada, para ello simplemente compararemos si el comportamiento esperado del análisis del problema se ha implementado de forma adecuada.

Ejercicio 1: Contrarreloj

Descripción: Crear un temporizador simple en el que el usuario pueda introducir una cantidad de segundos, y al hacer clic en un botón, comience la cuenta regresiva. Cuando el tiempo llegue a cero, debe mostrar un mensaje que diga "¡Tiempo finalizado!".

- **HTML:** Un campo de entrada (input) para ingresar los segundos y un botón para iniciar la cuenta regresiva. Un área para mostrar el tiempo restante.
- **JS:** Detectar el evento click en el botón, iniciar la cuenta regresiva y actualizar el tiempo en la página cada segundo hasta llegar a cero.

Objetivos:

- Uso de eventos (click).
- Métodos de temporización como setInterval y clearInterval.
- Manipulación básica del DOM para mostrar el tiempo restante.

Pruebas: Vídeos GIF en Readme.md en el que se muestre lo siguiente:

1. Rellenar el campo de tiempo con el valor 2 segundos y pulsar el botón "Iniciar", verificando que el temporizador cuenta correctamente y muestra una alerta al terminar los 2 segundos.
2. Repetir la prueba con el valor 4 segundos, asegurándose de que el temporizador cuenta hasta los 4 segundos y que al finalizar se muestra la alerta correspondiente.

Ejercicio 2: Calculadora básica

Descripción: Crear una calculadora básica que permita realizar operaciones de suma, resta, multiplicación y división. Los usuarios deben poder introducir dos números y seleccionar una operación. Al hacer clic en "Calcular", se debe mostrar el resultado de la operación seleccionada.

- **HTML:** Dos campos para introducir los números, una lista de selección (select) para elegir la operación, un botón "Calcular" y un área para mostrar el resultado.
- **JS:** Detectar el evento click en el botón "Calcular", capturar los valores de los campos y realizar la operación matemática seleccionada. Mostrar el resultado en la página.

Objetivos:

- Manejo de eventos (click).
- Validación de entradas de usuario (asegurarse de que los campos no estén vacíos).
- Uso de métodos matemáticos básicos (+, -, *, /).

Pruebas: Vídeos GIF en Readme.md en el que se muestre lo siguiente:

1. Probar que la suma de $10 + 12$ muestra correctamente el resultado de 22.
2. Verificar que la resta de $10 - 12$ devuelve el valor esperado de -2.
3. Comprobar que la multiplicación de 5×4 es 20, y que 5×0 devuelve 0.
4. Probar que la división de $5/4$ es 1.25 y que $5/0$ devuelve el mensaje "INDEFINIDO".

Ejercicio 3: Encuesta con gráficos de resultados

Descripción: Crear una página con una pequeña encuesta donde el usuario puede seleccionar una opción entre varias (por ejemplo: "¿Cuál es tu color favorito?"). Al enviar la respuesta, se deben mostrar los resultados en un gráfico de barras simple que se actualice dinámicamente cada vez que se envía una respuesta.

- **HTML:** Pregunta con varias opciones (utilizando radio buttons), un botón "Enviar", y un área donde se mostrará el gráfico de barras con los resultados.
- **JS:** Detectar el evento click en el botón "Enviar", actualizar las respuestas seleccionadas y generar un gráfico de barras simple manipulando el DOM. El gráfico puede ser construido con divs o tablas que cambian su tamaño en función de los votos.

Objetivos:

- Manejo de eventos (click).
- Manipulación del DOM para actualizar el gráfico de barras dinámicamente.
- Lógica para contar votos y reflejar visualmente.

Pruebas: Vídeos GIF en Readme.md en el que se muestre lo siguiente:

1. Verificar que al seleccionar una opción y hacer clic en "Enviar", se actualicen correctamente los votos en el gráfico.
2. Comprobar que el gráfico de barras se actualiza dinámicamente después de cada nueva votación.
3. Probar que el sistema gestiona correctamente intentos de envío sin seleccionar ninguna opción.
4. Verificar que las votaciones previas se mantienen después de varias interacciones.

Ejercicio 4: Cronómetro con controles de inicio, pausa y reinicio

Descripción: Crear un cronómetro que permita al usuario iniciarlo, pausarlo y reiniciarlo. El cronómetro debe actualizarse cada segundo y mostrar el tiempo en minutos y segundos.

- **HTML:** Botones para "Iniciar", "Pausar" y "Reiniciar", y un área para mostrar el tiempo.
- **JS:** Detectar los eventos click en los botones para iniciar el cronómetro, detenerlo y reiniciarlo. Usar `setInterval` para actualizar el cronómetro cada segundo.

Objetivos:

- Uso avanzado de eventos (click en múltiples botones).
- Métodos como `setInterval`, `clearInterval`.
- Manipulación continua del DOM (actualización del tiempo).

Pruebas: Vídeos GIF en `Readme.md` en el que se muestre lo siguiente:

1. Probar que el cronómetro comienza a contar correctamente al hacer clic en "Iniciar".
2. Comprobar que al pausar el cronómetro se detiene en el tiempo actual y no continúa avanzando.
3. Verificar que al hacer clic en "Reiniciar", el cronómetro vuelve a cero.
4. Probar que el cronómetro maneja correctamente largos períodos de tiempo sin errores.

Ejercicio 5: Juego de memoria con cartas

Descripción: Crear un juego de memoria donde el usuario debe hacer clic en dos cartas para ver si coinciden. Si las cartas coinciden, permanecen boca arriba, si no, se voltean nuevamente después de un breve intervalo. El objetivo es encontrar todas las parejas de cartas.

- **HTML:** Crear una cuadrícula de cartas (pueden ser imágenes o solo texto, como números o letras).
- **JS:** Detectar el evento click en cada carta. Usar lógica para manejar la comparación de cartas seleccionadas, y un temporizador para voltearlas si no coinciden. Al final, mostrar un mensaje cuando el jugador haya encontrado a todas las parejas.

Objetivos:

- Uso avanzado de eventos (click en elementos dinámicos).
- Uso de métodos de temporización como `setTimeout` para la lógica de volteo.
- Lógica condicional compleja para comparar cartas y gestionar el estado del juego.
- Manipulación dinámica del DOM (mostrar/ocultar cartas y actualización del estado del juego).

Pruebas: Vídeos GIF en `Readme.md` en el que se muestre lo siguiente:

1. Verificar que las cartas se voltean al hacer clic y se quedan volteadas si coinciden.
2. Comprobar que dos cartas no coincidentes se voltean de nuevo después de un breve intervalo.
3. Probar que el juego detecta cuando se han encontrado todas las parejas y muestra un mensaje de finalización.
4. Verificar que el juego puede reiniciarse correctamente barajando las cartas de nuevo.

Criterios en el formato de entrega

Todos estos ejercicios se deberán entregar en el formato establecido en clase o tablón de classroom, respetando las horas de entrega de cada uno de ellos indicados en la tarea de classroom.

¿Qué y cómo se entrega?

- Hay que realizar cada apartado de ejercicios en HTML diferentes y subirlo al repositorio a la carpeta SPRINT 2
- Hay que realizar un vídeo en formato .gif para cada ejercicio en el que se interactúe de manera dinámica con la web y adjuntarlo en el README.md del repositorio GIT en la carpeta del sprint correspondiente.
- Hay que realizar una captura de pantalla de aquellos ejercicios que sean estáticos y adjuntarlos en el README.md del repositorio GIT en la carpeta del sprint correspondiente.
- La estructura de carpetas del repositorio de GitHub debe seguir las normas establecidas de "EC" -> "T1" -> "SPRINTX", siendo X en la palabra SPRINTX el correspondiente al sprint actual.
- Los GIF se tiene que ver dentro de la previsualización del README.md.
- Para los boletines de sólo HTML y Javascript NO se podrán desplegar, sólo podrán consultarse mediante el .html abriéndolo el archivo con el navegador.
- Ejemplo de un README a continuación:

ANGULAR

BOLETÍN A1 AVANZADO 📄

Análisis del problema.

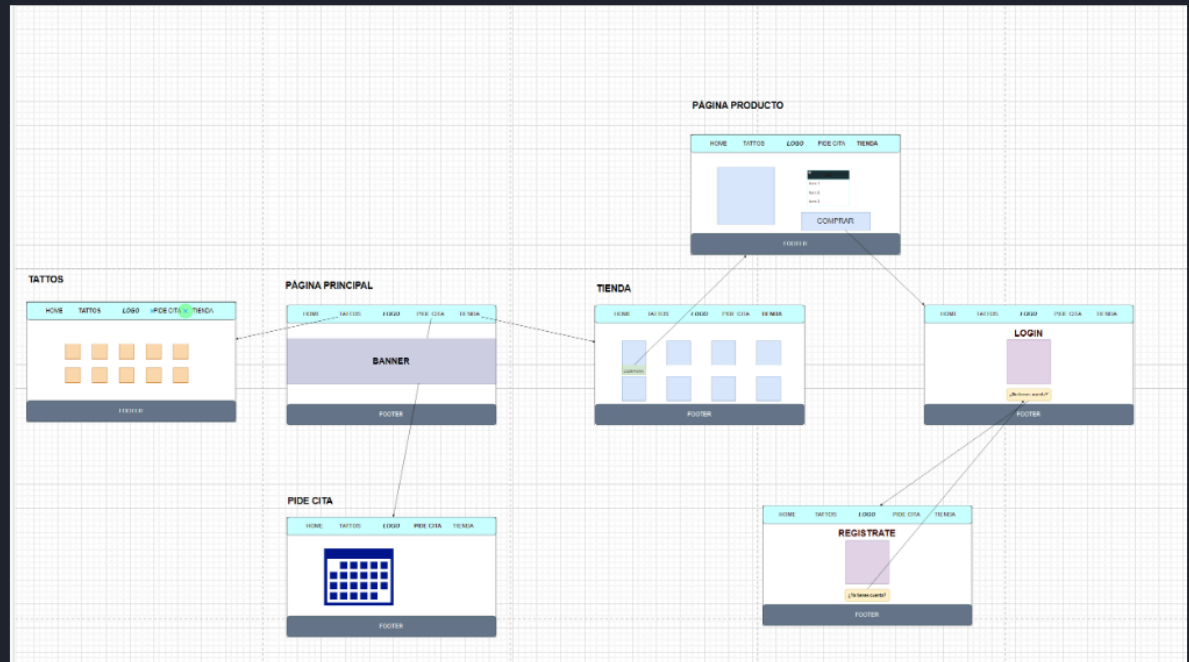
-> Se requiere realizar los siguientes ejercicios:

1. Seguir los pasos de instalación del vídeo suministrado "Instalación Angular.mp4".
2. Crear los componentes que se indican en la siguiente imagen con la ayuda del vídeo "Componentes Angular.mp4".
3. Añadir las verificaciones de cada input-botón mostrando un error en caso de que no se cumplan:
 - Email: Debe ser un email con su @ y otras validaciones. Buscar en internet. No debe estar vacío.
 - Password: No deben mostrarse los caracteres, sólo los puntos. No debe estar vacío.
 - Last Name: Puede estar vacío. Es un input normal.
 - First Name: No puede estar vacío. Es un input normal.
 - Botón con texto: Puede estar habilitado o deshabilitado.
 - Hiperenlace: Mostrará diferentes textos.
4. Añadir un componente que se pueda utilizar para el proyecto individual.
5. Realizar un boceto de cada una de las pantallas que se vayan a utilizar en el proyecto individual con la finalidad de reutilizar este trabajo y así reducir el esfuerzo total.

Primero instalaremos Angular con las indicaciones del profesor y con ayuda del vídeo. Luego creamos los componentes (Login, registro y footer).

Implementación de la solución.


En este apartado vamos a ponernos a implementar todos los apartados anteriores, vamos a hacer el ejercicio completo y los gifs de cada prueba.

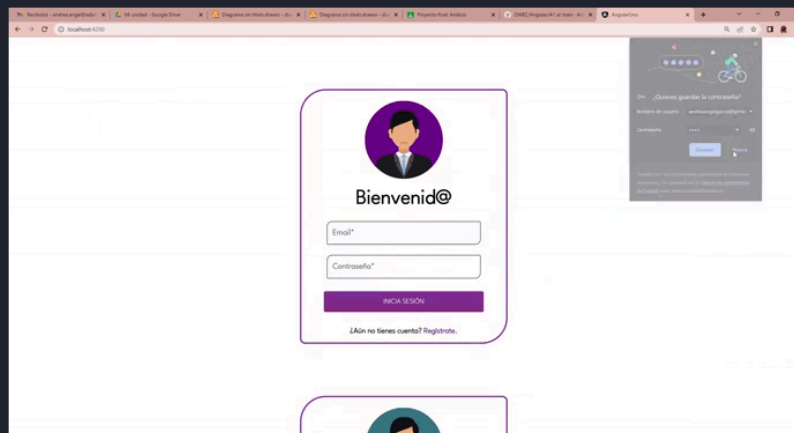


Pruebas.

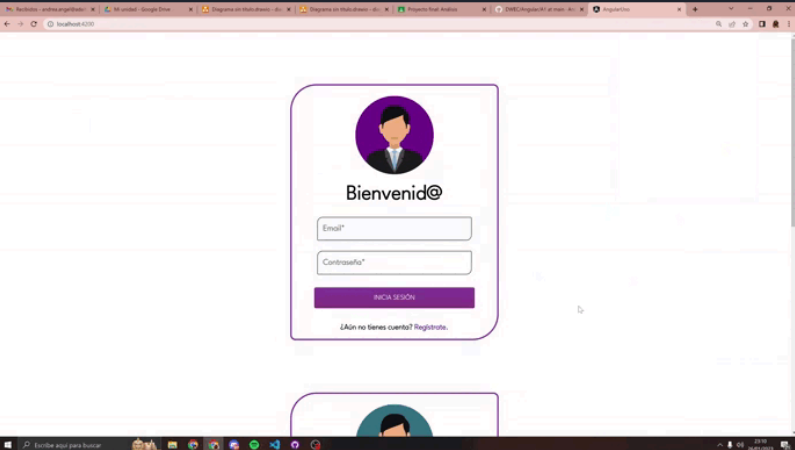
-> Plan de pruebas:

TestID	TestName	Description	StepNumber	StepAction	StepCondition
1	CASO OK - LOGIN	Entramos en la página, escribimos el correo y la contraseña correctamente y nos deja darle al botón.	1	Abrimos la URL.	Aparece una página web con login, registro y footer
			2	Escribimos el correo correctamente	Se muestra el correo
			3	Escribimos la contraseña	Aparece la contraseña oculta y podemos darle al botón
2	CASO KO1 - LOGIN	Entramos en la página, escribimos el correo y no ponemos ninguna contraseña.	1	Abrimos la URL.	Aparece una página web con su logo, buscador y una tabla
			2	Escribimos el correo correctamente	Se muestra la tabla vacía y el mensaje de que no hay ningún pokémon
			3	No escribimos ninguna contraseña	No nos deja darle al botón
3	CASO KO2 - LOGIN	Entramos en la página, escribimos mal el correo y no nos deja darle al botón.	1	Abrimos la URL.	Aparece una página web con su logo, buscador y una tabla
			2	Escribimos mal el correo	Se muestra la tabla vacía y el mensaje de que no hay ningún pokémon
			3	No escribimos ninguna contraseña	No nos deja darle al botón
4	CASO OK - REGISTRO	Entramos en la página, escribimos los datos correctamente y nos deja darle al botón.	1	Abrimos la URL.	Aparece una página web con login, registro y footer
			2	Escribimos el correo correctamente	Se muestra el correo
			3	Escribimos la contraseña	Aparece la contraseña oculta y podemos darle al botón
5	CASO KO - REGISTRO	Entramos en la página, escribimos los datos incorrectamente y no nos deja darle al botón.	1	Abrimos la URL.	Aparece una página web con login, registro y footer
			2	Escribimos los datos incorrectamente	No nos deja darle click
			3	No nos deja darle al botón	Aparece la contraseña oculta y no podemos darle al botón

-> LOGIN: 

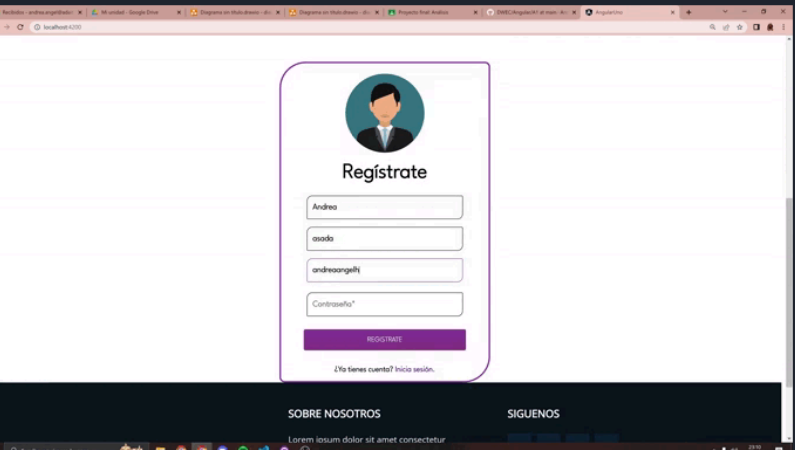


-> LOGIN: [🔗](#)



CASO OK:
Entramos en la página,
ponemos el correo correctamente
e introducimos la contraseña.

-> REGISTRO: [🔗](#)



CASO OK:
Entramos en la página,
ponemos los datos correctamente
e introducimos la contraseña.

Para los que quieran poner vídeos de la ejecución:

Convertidor Vídeo a GIF

- Hacer click a *Elegir Archivo* y elige el vídeo.
- Una vez elegido, click en *Upload video!*
- Seleccionar *Convert*.
- Puedes recortar el GIF para hacerlo más corto.

Video to animated GIF converter (ezgif.com)

Criterios de evaluación

Ejercicio 1	No se cumple con alguno de los criterios del formato de entrega o no se realizan las pruebas o no se realiza el gif que verifique el correcto funcionamiento del ejercicio. 0 puntos	Se cumple con lo establecido en los criterios del formato de entrega, se realizan las pruebas con éxito y se evidencia mediante un gif. Se encuentra todo documentado en el README.md 1 punto
Ejercicio 2	No se cumple con alguno de los criterios del formato de entrega o no se realizan las pruebas o no se realiza el gif que verifique el correcto funcionamiento del ejercicio. 0 puntos	Se cumple con lo establecido en los criterios del formato de entrega, se realizan las pruebas con éxito y se evidencia mediante un gif. Se encuentra todo documentado en el README.md 1 punto
Ejercicio 3	No se cumple con alguno de los criterios del formato de entrega o no se realizan las pruebas o no se realiza el gif que verifique el correcto funcionamiento del ejercicio. 0 puntos	Se cumple con lo establecido en los criterios del formato de entrega, se realizan las pruebas con éxito y se evidencia mediante un gif. Se encuentra todo documentado en el README.md 2 puntos
Ejercicio 4	No se cumple con alguno de los criterios del formato de entrega o no se realizan las pruebas o no se realiza el gif que verifique el correcto funcionamiento del ejercicio. 0 puntos	Se cumple con lo establecido en los criterios del formato de entrega, se realizan las pruebas con éxito y se evidencia mediante un gif. Se encuentra todo documentado en el README.md 2 puntos
Ejercicio 5	No se cumple con alguno de los criterios del formato de entrega o no se realizan las pruebas o no se realiza el gif que verifique el correcto funcionamiento del ejercicio. 0 puntos	Se cumple con lo establecido en los criterios del formato de entrega, se realizan las pruebas con éxito y se evidencia mediante un gif. Se encuentra todo documentado en el README.md 4 puntos