AJAX- CSS – JS - Bootstrap – Web Reactiva

Objetivos

El objetivo principal de esta práctica es afianzar los conceptos en la programación de hojas de estilo como así también poder integrarlas con el uso del lenguaje JavaScript y HTML Dinámico para realizar Paginas Reactivas.

Además ver los fundamentos de JavaScript e incorporar frameworks de desarrollo front end.

Herramientas / Preparación

- Para alcanzar este objetivo se podrá recurrir a las referencias bibliográficas presentadas en la teoría en clase.
- El trabajo debe implementarse con cualquier cualquier editor de texto plano o hacer uso de algunas de las herramientas recomendadas en clase como ser:
  - Visual Code, Adobe Dreamweaver, NetBeans, Sublime Text, entre otros.
- Framework
  - Bootstrap https://getbootstrap.com/
  - Materialize https://materializecss.com/

Organización

Grupal de 2 a 3 personas.

Fecha de entrega / presentación

Solo se debe presentar el ejercicio 13 y 14 de manera grupal.
Formato de presentación: Digital (Exposición Grupal).
Fecha tope de entrega: **10/10/2024 (Fecha sujeta a revisión)**
Se entregan las tres actividades

Ejercicios

Actividad 1 Adivinar el Número

Objetivo
En esta actividad, los estudiantes aprenderán a utilizar acciones repetitivas de JavaScript – Ajax para crear un juego sencillo de adivinar el número.

Descripción
El juego consiste en que el programa genera un número aleatorio entre 1 y 1000, y el usuario tiene que adivinar ese número. El programa le irá indicando al usuario si el número que ingresó es mayor o menor que el número secreto, y el usuario tendrá que seguir intentando hasta que acierte.

Instrucciones
1. Pide al usuario que ingrese un número entre 1 y 1000.
2. Genera un número aleatorio entre 1 y 1000 usando `Math.random()`.
3. Compara el número ingresado por el usuario con el número aleatorio generado.
4. Si el número ingresado es menor que el número secreto, indica al usuario que el número es muy bajo.
5. Si el número ingresado es mayor que el número secreto, indica al usuario que el número es muy alto.
6. Si el número ingresado es igual al número secreto, felicita al usuario por haber adivinado el número con un efecto visual llamativo
7. Repite los pasos 1-6 hasta que el usuario adivine el número secreto.

Consideraciones del sistema
- Mostrar al usuario cuántos intentos lleva en total en la partida actual.
-Agregar Opción para reiniciar partida actual (solo reinicia la cantidad de intentos y en número secreto).
- Se deberá guardar el mejor puntaje (menor número de intentos) y mostrarlo al final o en alguna parte del mismo.
-deberá mostrar la cantidad de partidas finalizadas y el promedio de intentos de ellas durante todo el juego.
-Todos los cambios de estado se deben realizar en tiempo real a medida que el usuario hace uso del juego (ejemplo: si el usuario arriesga se debe actualizar el número de intentos de manera síncrona. Si el usuario acierta se debe actualizar la cantidad de partidas finalizadas y el promedio)
- Los aspectos visuales del juego queda a criterio de desarrollo, pero contemplando los conceptos vistos en la materia.
-- Puede agregar otras funcionalidades adicionales al juego además de las mencionadas.

Actividad 2: Selector de Número Aleatorio (lotería o turnero aleatoreo)

Objetivo

En esta actividad, los estudiantes aprenderán a trabajar con JavaScript – Ajax para manipular un Json para crear un turnero que genera los turnos en forma Random.

Descripción:

Armar un turnero aleatorio, que permita al usuario por medio de un evento mostrar el número resultante en forma random. Se puede consultar el sitio http://random.org como modelo de generación.

Consideraciones:

a. El sistema debe prever que el usuario pueda colocar los límites del random EJ de 10 a 60. Estos datos deben ser almacenados en algún archivo de texto para cuando se ingrese al sistema nuevamente ya están seteados.
b. A diferencia del sitio modelo **http://random.org**, el sistema debe ir guardando los números resultantes para evitar mostrar números repetidos. Para el almacenado puede utilizar un archivo en cual guarde los números que han salido.
c. En caso de que el número generado este ya en el medio de almacenamiento, el sistema debe brindar un número disponible no repetido, para esto se puede mostrar el número superior, si no existe deberá elegir el número inmediatamente inferior. Si no hay más números mande un mensaje "todos los números están generados".
d. Debe permitir reiniciar el sistema, es decir blanquear tanto los índices random y los números ya salientes.
e. Para el guardado de datos puede usar el siguiente formato de Json (puede usar otro medio de guardado que usted considere conveniente en para los números. Un ejemplo válido de ejemplo es el siguiente Json:

```
{
    "limite inferior": 50,
    "limite superior": 40,
    "números": [
        {
            "numero": 2
        },
        {
            "numero": 9
        },
        {
            "numero": 3
        }
    ]
}
```

Actividad 3: Manejo de Api

Objetivo:
Utilizando una api online ejemplo elabore una web reactiva con las acciones típicas de un crud.
Dándole un formato y estilo profesional utilizando los frameworks css vistos.

Herramientas para el Ejercicio

- APIs sugeridas
  - https://jsonplaceholder.typicode.com/
  - https://zudoku.dev/demo?api-url=https://zudoku.dev/petstore.oas.json
  - https://app.mockfly.dev/ (pueden crear su propia api)
  - o pueden utilizar la api que están desarrollando en la materia de Análisis

  .
- Framework
  - Bootstrap https://getbootstrap.com/
  - Materialize https://materializecss.com/

Consideraciones:
Realice una web reactiva que consume la api seleccionada y que contenga las siguientes acciones

- listar entidades
- permitir ordenar el listado por diferentes campos
- filtrar el listado de entidades por diferentes filtros
- permitir paginar el listado de entidades
- dar de alta una entidad
- editar una entidad
- borrar una entidad
- buscar una entidad