

Sitio Web Duelos

Una compañía de software que está emprendiendo ha contactado a su equipo para desarrollar un sitio web donde los usuarios puedan tener duelos entre sí. La compañía requiere hacer una licitación para elegir entre las propuestas presentadas por varios equipos y desarrolladores. La compañía tiene la idea probar con un un sitio web sobre duelos entre usuarios, pero no sabe qué tipo de duelos, ya que más bien ahí es donde desean conocer la creatividad y el potencial de los equipos que estaría contratando para un proyecto muy grande que pondrá a la nueva compañía en la palestra.

Objetivo

Desarrollar una página web que permita a los jugadores participar en duelos, donde se carguen equipos y estadísticas de los usuarios, se asignen roles, jugar contra "invitados" (no se cargan estadísticas, ni se guardan al final del duelo).

Características y Funcionalidades

1. Autenticación de Usuarios:

- Inicio de sesión con correo electrónico y contraseña.
- Registro de usuario mediante correo electrónico y contraseña.

2. Gestión de Amigos:

- Posibilidad de agregar amigos por medio de correo electrónico o nombre de usuario.

3. Configuración de Equipos:

- Posibilidad de armar equipos, con opción de guardarlos en su cuenta para ser utilizados desde cualquier lugar.

4. Historial y Estadísticas:

- Cálculo y visualización del promedio de victorias.
- Mostrar estadísticas similares para cada amigo agregado.
- Registro de todos los duelos realizados y con cuál usuario.
- Promedio de victorias y derrotas por elemento (ver sección de ideas de duelos).
- En el perfil del usuario debe mostrarse:
 - La lista de los elementos individuales más utilizados, su promedio de victorias y derrotas, el número total de veces que ha sido utilizado, con cuáles elementos lo han derrotado más y a cuál le ha ganado más.
 - Nombre de usuario (no es lo mismo que el correo pues esto es un dato privado).
 - Posibilidad de cambiar la contraseña.

- Posibilidad de cambiar la foto de perfil.
- Lista de amigos.
- Top 3 equipos con más victorias.

5. Integración con API:

- Utilización de datos utilizando estilo API para calcular daños, vida, movimientos, ítems y sus efectos durante los duelos.

Tecnologías

- **Front-end:** Deberá desarrollar el sitio web utilizando únicamente HTML, CSS y JavaScript puro, sin frameworks o librerías de estilos como Bootstrap, JQuery, React, Vue, Next, Tailwind, etc.

- **Back-end:** Los estudiantes indicarán al profesor cuanto antes cuál lenguaje de servidor utilizarán. De este lado tampoco se permitirá el uso de frameworks avanzados más allá de las herramientas básicas web (en caso de duda preguntar). Entre las tecnologías disponibles para backend puede elegir alguna de las siguientes opciones:

1. Express.js (NodeJS)
2. Flask (Python)
3. JSP/Servlets (Java)
4. PHP (PHP)
5. ASP.NET (C#)
6. ¿Piensa en otro? Consultar con el profesor.

- **DataBase:** MongoDB.

Listado de páginas (mínimo)

1. Inicio de sesión.
2. Registro de usuario.
3. Recuperación de contraseña.
4. Verificación de la cuenta.
5. Listado de equipos.
6. Creación de equipos y escogencia de habilidades/movimientos de cada elemento.
7. Edición de equipos y escogencia de habilidades/movimientos de cada elemento.
8. Manejo o búsqueda de amigos.
9. Historial de partidas/duelos.
10. Perfil del usuario.
11. Perfil de otros usuarios.
12. Tablas de liderazgo (lista de usuarios con más victorias).
13. Pantalla para prohibir/habilitar el uso de ciertos elementos.
14. Agregue más de acuerdo a sus análisis.

Lógica de los duelos

Debe quedar claro que no estamos construyendo un videojuego online, donde se tenga que administrar la interacción de los jugadores en tiempo real. El tipo de duelo o enfrentamiento que se solicita es más bien un evento "pasivo". Es decir, en el momento de un duelo ambos usuarios enviarán sus movimientos a un servicio central. El eje central recibe las dos partidas de movimientos y ejecuta la lógica del enfrentamiento definida para determinar resultados, los cuales serán notificados a los jugadores.

Entre las ideas disponibles para duelos puede elegir alguna de las siguientes ideas:

1. Duelo de cartas.
2. Enfrentamiento de bichos (ej. pokemones).
3. Piedra, papel y tijeras.
4. Gato (tic-tac-toe), 4 en raya, etc.
5. Desafíos de preguntas/respuestas. Adivinanzas.
6. Penalitos.
7. ¿Piensa en otro? Consultar con el profesor

Cada equipo deberá indicarle al profesor cuanto antes cuál idea desean trabajar, ya que todos los equipos deberán trabajar en ideas distintas. Toda la creatividad y demás detalles que surjan a raíz de estas ideas corre por cuenta de los equipos.

Iteración 1: Análisis

Esta etapa debe realizarse y entregarse de forma individual, aunque de algún modo la trabajen de forma grupal.

Entregables:

1. Preguntas y/o dudas del requerimiento.
2. Modelado de datos (entidades y atributos), representado en formato JSON.
3. Mapa del sitio (cualquier diseño: árbol jerárquico, araña, etc)
4. Prototipos de pantallas (wireframes).

Configuración: Un representante de cómo serán las pantallas de configuración o mantenimiento de datos. Estas pantallas es donde se lleva a cabo el *CRUD* de las entidades.

Transacciones: Todas las pantallas referentes al duelo; éstas incluyen la invitación, la aceptación, el duelo, los resultados.

Reportes: Un representante de cómo serán las pantallas de reportes. Puede usar una representación tabular de la información, o también puede presentar gráficos de barras, de *pie*, de puntos, etc). Deben haber filtros para personalizar los reportes.

IMPORTANTE: Deberá escribirlo y dibujarlo todo a mano. Todo, tanto el JSON, como el *sitemap* deberá escribirlo con el mayor orden y la mejor caligrafía, ya que este aspecto será parte de esta evaluación. Para el caso de los prototipos deberá hacer lo que haría en una herramienta como Figma, pero a mano, y de paso explota ese talento de dibujante que posiblemente no sabía que tenía escondido. Utilice útiles escolares como reglas y lápices de color y haga que sus diseños tengan un matiz artístico.

Iteración 2: Front-End

Esta etapa y la siguiente deben realizarse y entregarse de forma grupal.

Entregables:

1. Preguntas y/o dudas del requerimiento.
2. Sitemap (correcciones del análisis 1)
3. JSON Database (correcciones del análisis 1)
4. HTML, CSS y JS-DOM.
5. Todas las pantallas deben estar visualmente funcionales.
6. Validaciones de formularios (marcas de error y alertas).
7. Notificaciones de éxito/error por acciones de usuario.

Esta entrega y la siguiente la realizan por medio del Git de la U y de Mediación Virtual. Posteriormente se brindarán más detalles.

Iteración 3: Back-End

Entregables:

1. Sitemap (final)
2. JSON Database (final)
3. HTML, CSS y JS-DOM (correcciones iteración 2).
4. JS-Server (fetch API)
5. Back-end API Controller (código fuente del backend)
6. MongoDB API Model (código fuente de la interacción con BD)