



El proyecto se basa en el juego de móvil de Dragon Ball: Dokkan Battle. Se trata de una página en la que cada usuario registrado podrá coleccionar una serie de cartas de distintos personajes de Dragon Ball. Estas cartas se conseguirán en una sección de Gachapón.

REQUISITOS FUNCIONALES

Nuestro proyecto tiene los siguientes requisitos funcionales que cumplirán con los objetivos y las necesidades de este:

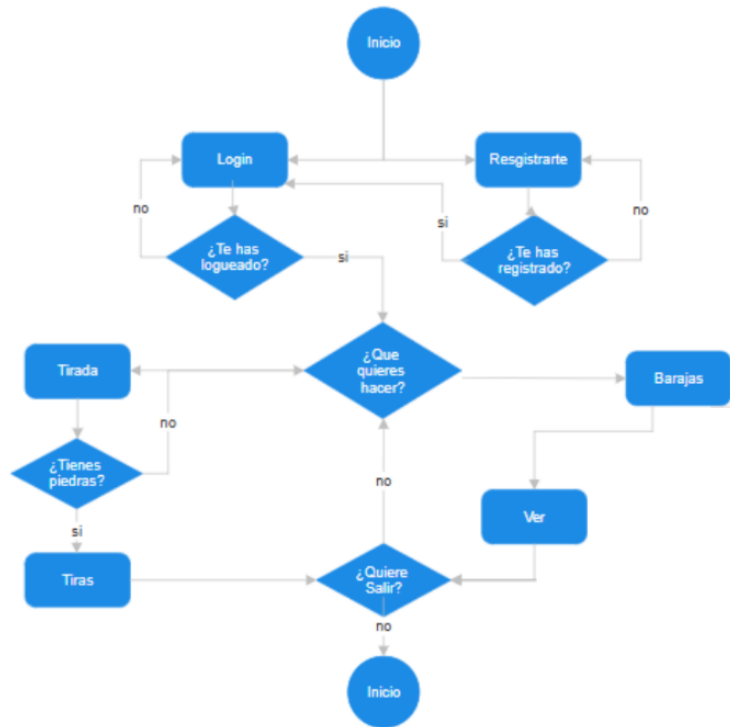
- **Gestión de usuarios:**
Cada usuario podrá crear una cuenta, iniciar sesión y visualizar su cuenta y/o modificarla.
- **Colección de personajes:**
Cada usuario tendrá un inventario propio donde irá almacenando los personajes conseguidos del gachapón.
- **Gachapón:**
Cada usuario podrá entrar en la sección de Gachapón para poder conseguir personajes.
- **Piedras:**
Cada usuario podrá conseguir piedras mediante un clicker. Estas piedras se usarán para tirar a personajes.
- **Actualización continua:**
A medida que avance el proyecto, se podrá ir actualizando y añadir nuevo contenido como personajes, funcionalidades, etc.

REQUISITOS DE INTERFAZ

1. Identificación de Usuarios y Casos de Uso:

La aplicación la usarán sólo los usuarios que se registren en el login, el objetivo es completar una colección, el usuario inicia sesión y entra en su usuario y ya dentro puede conseguir personajes.

En nuestro diagrama de flujo se representa el funcionamiento que tendrá nuestro Dokkan Battle y qué opciones tienen los usuarios en él.



2.Requisitos de Usabilidad y Experiencia de Usuario:

Habrá un inicio con unos botones para poder registrarse o iniciar sesión.

Una vez iniciado sesión el usuario tendrá una interfaz con varios botones para hacer diferentes acciones, por ejemplo, ver el inventario, conseguir piedras o conseguir personajes nuevos.

La página estará ambientada en Dragon Ball, por lo que usaremos colores cálidos y de la paleta de color de Dragon Ball.

3.Definición de Componentes de la Interfaz:

- Pantallas principales: Inicio, Login/Registro, Perfil, Inventario, Quienes somos, Gachapón y Conseguir Piedras.
- Elementos UI: Botones, alertas, menús desplegables y cards.
- Animaciones y transiciones: Uso de imágenes para las cartas, animaciones en toda la interfaz, transiciones para el gachapón, etc.

4.Herramientas usadas en el desarrollo de la interfaz:

VisualStudio(Programación), Trello(Organización), GitHub(Control de versiones y trabajo en equipo), Canva(Diseño de interfaz), XAMPP(Servidor Local), Draw.io(Diagramas), Postman (manejo de APIs).

5.Estándares Técnicos

- Lenguajes de programación:
 - Front-End: HTML, CSS
 - Frameworks: Bootstrap
- Tendrá compatibilidad con distintos navegadores.

PLANTILLA DE REFERENCIA PARA LA INTERFAZ

The wireframe illustrates a web interface layout. The top section features a light gray header with 'REGISTRARSE' and 'LOGIN' links on the right. Below the header, the main content area contains a large 'LOGO' placeholder, a 'Pequeña descripción' (Small description), a blue button labeled 'QUIENES SOMOS' (Who we are), and a faint 'INICIO' (Home) link. A note 'VIDEO DE FONDO DOKKAN' (Dokkan background video) is positioned at the bottom of this section. The bottom section contains a blue button labeled 'VOLVER' (Return) on the left. In the center, there is a blue registration form box with a 'LOGO' placeholder, input fields for 'Nombre' (Name), 'Correo electrónico' (Email), 'Nombre de usuario' (Username), 'Contraseña' (Password), and 'Tipo de usuario' (User type), and a 'REGISTRARSE' (Register) button. A faint 'REGISTRO' (Registration) label is visible on the right side of this section.

VOLVER

LOGO

Nombre de usuario

Contraseña

INICIAR SESIÓN

INICIO DE SESIÓN

BOTON LOGO

PIEDRAS

CONSEGUIR PIEDRAS

INVENTARIO

GACHAPÓN

PERFIL

TITULO

Pequeña descripción

QUIENES SOMOS

INICIO AL TENER SESION INICIADA

VIDEO DE FONDO DOKKAN

BOTON LOGO

PIEDRAS

CONSEGUIR PIEDRAS

INVENTARIO

GACHAPÓN

PERFIL

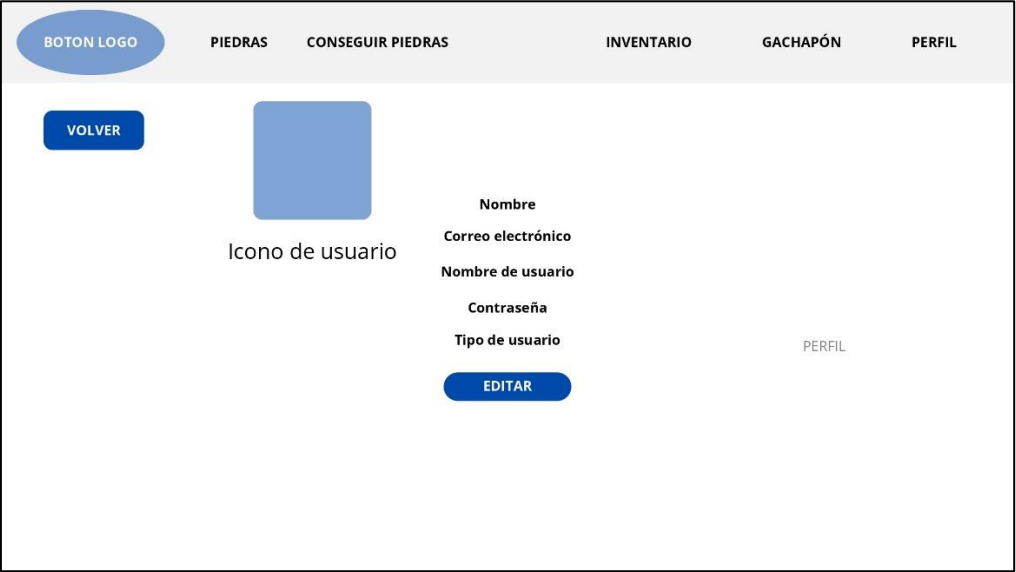
IMAGEN GACHAPÓN

TITULO GACHAPÓN

DESCRIPCIÓN

TIRAR

GACHAPÓN



REQUISITOS DE SISTEMA

1. Alojamiento Web:

Se utilizará un entorno de desarrollo local mediante XAMPP para el alojamiento de nuestro proyecto.

2. Requisitos de Hardware y Software:

- Hardware: Se utilizará un equipo con especificaciones adecuadas para ejecutar Visual Studio y un servidor local (XAMPP).

Se necesitará conexión a internet estable para GitHub y realizar pruebas si es necesario.

- Software:

Sistemas operativos que se usarán en el desarrollo: Windows.

Servidor web: Apache (incluido en XAMPP).

Lenguajes de programación:

- Frontend: HTML, CSS.
- Backend: PHP.
- Base de datos: MySQL.

- Herramientas de desarrollo:

Visual Studio, XAMPP (para el servidor local), MySQL Workbench (para la gestión de la base de datos), Postman (para pruebas de AP).

3. Consideraciones Adicionales:

- Escalabilidad: Se podrá migrar a un entorno de producción con hosting y dominio reales.

- Seguridad: Implementación de validaciones en formularios, uso de sesiones para la gestión de usuarios, etc.

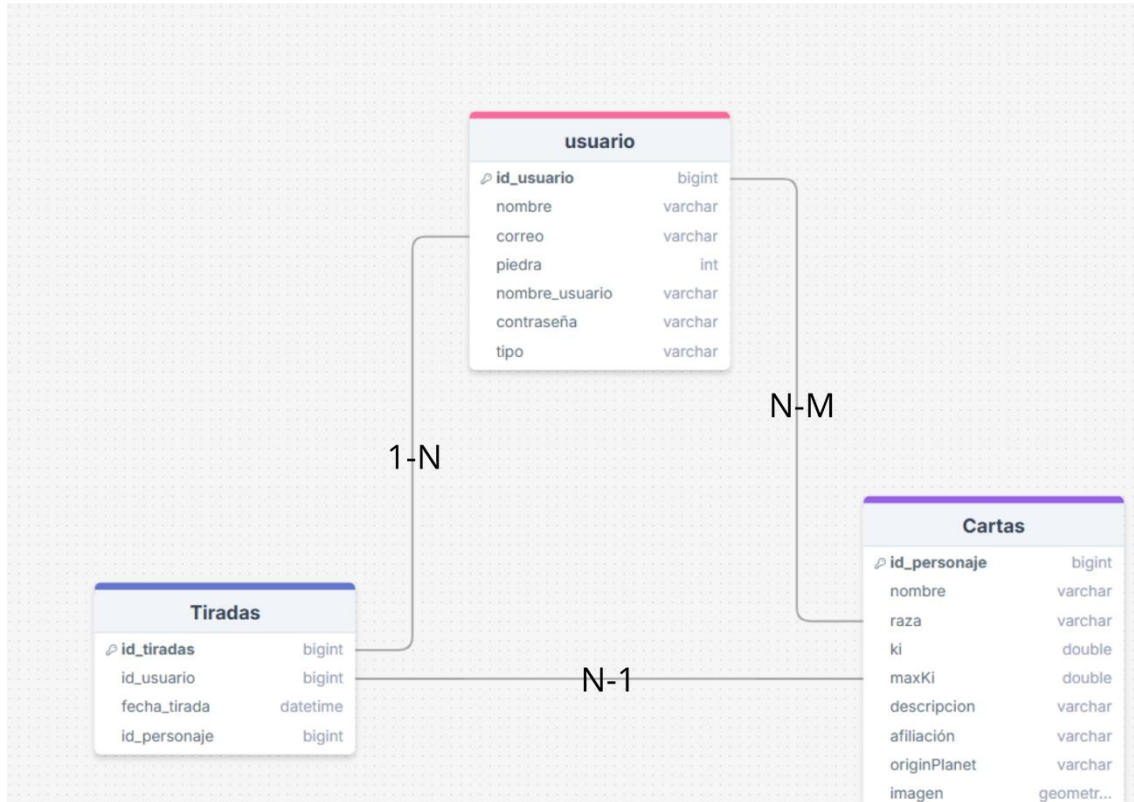
- Mantenimiento: Copias de seguridad, registro de errores y monitoreo del sistema mediante logs.

REQUISITOS DE ALMACENAMIENTO

1. Modelo de Datos y Estructuras:

-Tipo de datos: Usaremos datos estructurados para nuestra base de datos y nuestra aplicación

-Diseño de la Base de Datos:



En él podemos ver:

- Tabla Usuario:
 - o ID
 - o Nombre
 - o Correo electrónico
 - o Numero de piedras para el gachapón
 - o Nombre de usuario
 - o Contraseña
 - o Tipo de usuario (Admin o jugador)
- Tabla Carta:
 - o ID
 - o Nombre
 - o Raza
 - o Ki (habilidad)
 - o MaxKi (habilidad)
 - o Descripción de la carta
 - o Afiliación

- Planeta de origen
- Imagen del personaje
- Tabla Tiradas:
 - ID
 - ID del usuario
 - ID de la carta
 - Fecha de la tirada

Y las relaciones:

- Usuario – Carta:
Se trata de una relación N – M ya que un usuario puede tener muchas cartas y las cartas pueden tener muchos jugadores.
- Usuario – Tiradas:
Se trata de una relación 1 – N ya que un usuario puede hacer muchas tiradas, pero la tirada pertenece a un usuario.
- Tiradas – Carta:
Se trata de una relación N – 1 ya que una tirada puede contener una carta pero una carta puede estar en muchas tiradas distintas.

2. Tipo de Base de Datos:

Relacional (SQL): Usaremos phpMyAdmin para la gestión de la base de datos.

3. Rendimiento y Tiempos de Respuesta:

Se hará un código eficiente y sin necesidad de realizar funciones que no sean necesarias en determinados momentos.

4. Respaldo y Recuperación de datos:

Como se ha mencionado anteriormente, se utilizará GitHub para copias de seguridad y respaldo de versiones del proyecto en caso de que haya algún fallo.