

PokeSharp



Autor: Marcos Zahonero
Ciclo formativo: 2º DAM

Centro: IES Dr. Faustí Barberá
Tutor individual: Belén López Pérez



Índice

Introducción.....	3
Justificación.....	4
Tecnologías utilizadas.....	6
Juego.....	6
Web.....	6
Temporalización inicial.....	7
Temporalización final.....	8
Descripción del proyecto.....	9
Encuentro pokémon.....	9
Combate contra NPCs.....	13
Sistema de puertas.....	14
Pantalla menu principal.....	15
Pantalla Tus Pokémon (PC).....	16
Pantallas secundarias.....	18
Inicio de sesión.....	18
Menu pausa.....	20
NPCs.....	20
Entrenadores.....	20
Página web.....	20
Control de versiones.....	20
Discord.....	21
Diagrama Entidad-Relación.....	22
Esquema funcional.....	23
Dificultades encontradas y cómo las ha superado.....	24
Aprendizaje de nuevas tecnologías.....	24
Problemas con mayúsculas en los scripts.....	24
Ambición frente a limitaciones de tiempo.....	24
Diseño.....	25
Estructura carpetas.....	25
Implementación.....	26
Ideas a futuro.....	27
Experiencia personal y conclusiones finales.....	28
Referencias bibliográficas y webgrafía.....	29
Agradecimientos.....	30

Introducción

PokeSharp es un fan game de Pokémon basado en texturas del juego Pokémon Blanco y Negro 2. A diferencia de otros juegos fan games, este se diferencia sobre todo por sus tecnologías utilizadas, ya que la mayoría de juegos usan RPG Maker XP, el cual permite la facilidad de incorporar texturas y sistemas ya hechos, en este caso se partió desde cero, con el objetivo de crear una historia original y entretenida para el jugador.

Este proyecto incluye también una página web que sirve para su promoción, facilitando el seguimiento de las actualizaciones, el lanzamiento de tráilers y otros contenidos relacionados importantes del proyecto.

Justificación

El motivo de desarrollar este proyecto es principalmente personal, más que el realizar un proyecto complejo. Desde hace mucho sentía la necesidad de crear un juego, especialmente de Pokémon, que incluyera una historia única, con el objetivo de capturar el interés del jugador, haciendo que te sea agradable cada segundo que lo juegues.

Objetivos del proyecto

- Desarrollar un juego que mantenga la esencia de Pokémon, pero que al mismo tiempo contenga elementos diferenciadores.
- Implementar sistemas complejos (Combate, inventario, diálogos...) para poner a prueba mis conocimientos en programación.
- Explorar una nueva área, los videojuegos, y nuevas tecnologías, para así explorar más el mundo de la programación.
- Crear un proyecto que pueda ser continuado en un futuro y que pueda llegar a tener éxito a largo plazo.
- Realizar un proyecto con el que me sienta cómodo haciéndolo y satisfecho de lo conseguido.

Tecnologías utilizadas

Juego

C# = Lenguaje de programación principal.

Godot Engine = Motor de desarrollo del videojuego.

Supabase = Plataforma utilizada para conectividad y almacenamiento de bases de datos.

Web

HTML = Estructura base de la página web.

CSS = Diseño y estilo de la web.

JavaScript = Funcionalidades dinámicas, como por ejemplo traducción automática de idiomas y otras interacciones.

Temporalización inicial

Para los tiempos que debo seguir hice una hoja de cálculo para llevar una organización y porque también van a ser subidos a un canal de YouTube unos blogs de desarrollo, que son vídeos durante el desarrollo de un videojuego enseñando mecánicas y en sí el desarrollo del juego.

Fecha límite	26/02/25	23/03/25	06/04/25	20/04/25	04/05/25
Nº dev blog	1#	2#	3#	4#	5#
Nombre dev blog	Movement, hitboxes y entorno	NPCs in game, Dialogos y sonidos	Equipo, inventario y captura de pokemon	Combates e Items	Inicio, ajustes y sistema de guardado
Contenido en el video	Movement Grid System	NPCs in game	Encuentro de pokemon	Combate vs. NPCs	Pantalla inicio
	Y Sorting	Mapa spawn hecho	Ventana equipo	Cambio de mapa	Crear/Cargar partida
	Poder entrar en sitios	Sistema Dialogos	Ventana inventario	Recoger items	Crear personaje
	Ventana combate provisional	Hierba alta y sonidos		Menu	Ajustes
	Diseño provisional spawn				Sistema de guardado automatico

Antes de hacer eso también realicé la página web, que fue durante el curso hasta básicamente terminar el curso (Diciembre 2024 – Febrero 2025).

Para el mes de abril quisiera tener una posibilidad de otorgar una beta privada para si alguien le apetece probarlo y encontrar algunos fallos y también así ver un poco la evolución y opinión.

Temporalización final

La temporalización final que realicé fue bastante distinta a la temporalización inicial, esto se debe a la falta de experiencia con el entorno utilizado y también a la poca consideración de trabajo que llevaría hacer ciertas tareas.

Al final la idea de hacer los blogs de desarrollo fue cancelada por falta de tiempo y también porque se consideró que requería demasiado esfuerzo para el beneficio que aportaba realmente al proyecto.

Esto fue lo que se realizó en el plazo de los vídeos, y también lo que se mostró en ellos.

Fecha límite	06/04/25	13/04/25	20/04/25
Nº vídeo	1#	2#	3#
Contenido en el video	Pantalla inicio	Cambios en la estructura (Clases, carpetas, etc..)	Funcionalidad de ocultar opciones de combate
	Sistema movimiento del jugador	Navegación entre escenas	Mochila en combate
	Sistema de interiores para uso de puertas su sonido	Menú de pausa	Nuevo mapa
	Sistema de encuentros pokemon sin acabar	Estados de juego (Menú, Pausa, jugando, en combate)	Menú principal (Tu equipo, tus pokemon, mochila y pokedex)
	Vinculación de base de datos	Asignación de nivel a las hierbas para encuentros	

Después de estos vídeos planteo la siguiente temporalización final con lo que veía esencial, con fechas límite o fechas para empezar a realizar esa tarea.

- Pantalla de equipo → Día 04/05/25
- Música → Día 10/05/25
- NPCs → Día 15/05/25
- Inicio documentación → Día 09/05/25
- Lanzar versión alfa → Día 11/05/25
- Empezar Powerpoint → Día 16/05/25
- Sistema de encuentros pokémon → Día 17/05/25
- Pantalla de inventario → Día 18/05/25
- Sistema de combate contra NPCs → Día 24/05/25
- Pantalla de ajustes → Día 02/06/25

Descripción del proyecto

El proyecto se divide en sistemas que lo hacen un fan game de Pokémon, como por ejemplo los encuentros pokémon o los combates.

Encuentro pokémon

¿Cómo llegamos a un encuentro pokémon?

Primero debemos dirigirnos a buscar unas hierbas y pasar por encima, como vemos en la imagen:



Cuando estemos pasando por encima moviéndonos entre ellas, calcula internamente un número del 1 al 100. Si este número está entre 1 al 25, saltará el encuentro pokémon.

El procedimiento al iniciar el encuentro pokémon es el siguiente:

1. Salta la animación haciendo zoom, la música empieza y aparece la común animación gráfica de Pokémon en la pantalla.
2. Busca un aleatorio del 1 al 649 y consulta a la base de datos, en la tabla de pokemons, ese poke_id en busca de un pokémon.
3. Saca sus datos convirtiendo ese JSON a un objeto y empieza a asignar la parte gráfica, asignando el nivel del pokémon, nombre, etc.
4. Hace una búsqueda de sus movimientos. En este juego es distinto al juego original, ya que aquí existe una fórmula distinta con porcentajes, y es la siguiente:

- > 45 % -> Movimiento que puede aprender el Pokémon por tutor
- > 35 % -> Del mismo tipo del Pokémon
- > 20 % -> Movimientos aleatorios sin requisitos

Esto contiene unos porcentajes para decidir qué movimientos sacar. Esto lo decide con cada movimiento, es decir, un total de 4 veces.

5. Salta la pantalla de combate y aparece con una animación de negro a su color original.
6. Empieza realmente el combate y se empiezan a gestionar los estados, calculando el primer turno dependiendo la velocidad de ambos pokémon. Si uno es más veloz, empieza.

Aquí un ejemplo de un encuentro pokémon cuando ya ha cargado todo. En este caso, el juego consultó las dos velocidades y vio que en este caso el pokémon "Darkrai" era más veloz que Dialga. Es por eso que atacó primero, como se puede ver abajo a la izquierda, que es una caja que contiene el registro de combate, para también ver qué ataques usa el enemigo. Como se puede ver, al pokémon enemigo realizar ese ataque a mi pokémon, le ha bajado la vida en consecuencia.



Ahora que el enemigo ha atacado, se le acaba su turno y nos toca a nosotros. Podemos realizar cualquier acción: usar un movimiento, cambiar de pokémon o usar una Pokéball para atraparlo. Probemos primero con un movimiento:

Aquí vemos los movimientos. Su fondo depende del tipo que son: tres son de tipo dragón, por eso tienen ese color y, el Meteor mash es tipo acero.



Lo usamos y le bajamos la vida hasta bastante. Después cambia el turno y consigue atacarnos. Vamos a aprovechar que se encuentra a poca vida para poder ver la captura de pokémon.

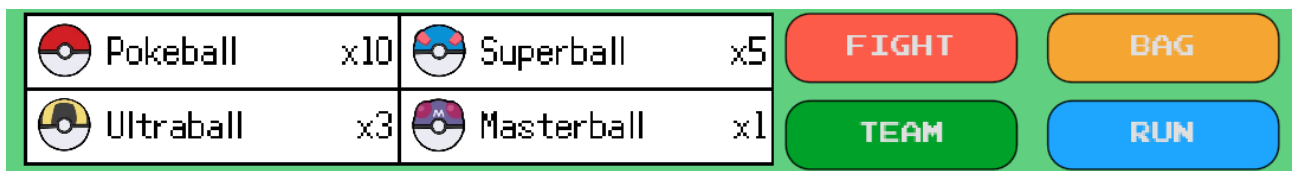


Cuando le damos a “Bag”, podemos usar objetos de captura.

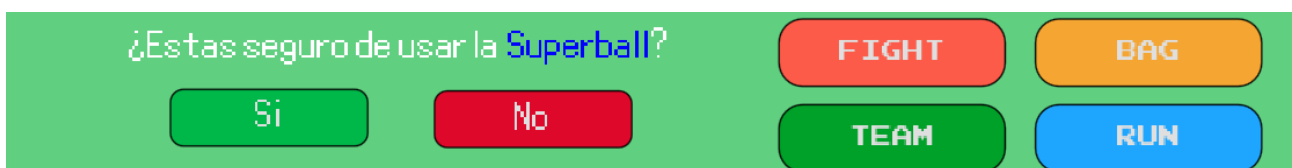


Vamos a usar una Superball, este pokémon que tenemos en contra es un pokémon legendario, es decir, tiene la probabilidad más baja de todas. Cada Pokeball que vemos aquí tiene un multiplicador propio:

- Pokeball x1
- Superball x1.5
- Ultraball x2
- Masterball captura asegurada.



Una confirmación, le damos a “Sí”.



No ha habido suerte y nos atacó de vuelta al acabar nuestro turno, ya que lo hemos utilizado para lanzar la Superball, vamos a probar con una Ultraball.



Tampoco, las probabilidades fueron altas, pero no fue suficiente. Al atacarnos, nos dejó sin vida restante, es decir, nos debilitó al Pokémon, ahora nos fuerza a cambiar de Pokémon mostrándonos lo siguiente:



Una vez seleccionemos uno, perdemos el turno y le toca atacar. Vamos a dar por finalizado este encuentro con una Masterball.

Sale que lo hemos capturado y finaliza el encuentro. Este pokémon se colocará dependiendo de nuestro equipo, si nuestro equipo actual se encuentra lleno se mandará al PC, sino se añadirá al equipo actual.



Combate contra NPCs

(Pendiente)

Sistema de puertas

Esta parte es de alta importancia a la larga en el proyecto, ya que este sistema permite el traslado del personaje entre distintas escenas/mapas sin hacer al usuario pensar que están cambiando muchas cosas de forma interna.

Al entrar a un interior, por ejemplo, al laboratorio, debemos crear en un json un nuevo registro con la siguiente estructura:

```
"Nombre": { "label": "Laboratorio", "interior": "false", "scene": "Escena que carga", "xSpawnPoint": "-84", "ySpawnPoint": "191,6667" }
```

Esta permite que cuando nos acerquemos a la puerta en este caso desplegué un texto encima de la puerta para hacernos saber donde vamos a entrar, llegados a este punto si le damos a la tecla "E" nos llevará dentro, haciendo sonar un sonido de una puerta, mostrando una transición y mientras la transición se realiza hace el cambio de escena sin que se vea visualmente.



Cuando ponemos si es un interior o no es para poder hacer ciertas cosas como lo siguiente, si nosotros entramos de espaldas al laboratorio ocurre algo curioso, y es que da igual como nos coloquemos que nos orienta hacia arriba.



Esto también ocurre al revés.



Pantalla menu principal

Esta pantalla se puede abrir mediante la tecla "R". A la parte izquierda tenemos la sección de **Equipo**, donde se muestra de forma visual los Pokémon que tienes en tu equipo. ¿Cómo identifica el juego qué pokémon tienes en tu equipo o no? ¿Y su orden? Esto es gracias a una variable que contiene los pokémon que capturas: inTeam. Es un entero que funciona como orden, y para identificar si está en tu equipo o si se encuentra en el PC es sencillo: si el número es -1, este lo enviará al PC. En el apartado de inicio de sesión se entrará más a profundidad sobre esto.

Por otro lado, en la parte de la derecha tenemos un menú con 2 cosas a futuro.

1. Tus Pokémon, esta pantalla nos abre la pantalla del PC, para poder gestionar los pokémon que tienes fuera del equipo y poder intercambiarlos para ponerlos en tu equipo.

2. Bolsa, es un apartado que contiene los objetos que posees. Esta opción es a futuro, es por eso que actualmente existe pero no tiene funcionalidad, es algo a futuro.

3. Pokedex, esto es algo común en todos los juegos Pokémon, pero es algo que cuesta bastante realizar y también es algo para la gente que de verdad quiera invertir más de 40 horas al juego. Ya que consiste en atrapar a todos los pokémon, eso quiere decir poder atrapar a los 649 que se encuentran actualmente en el juego, ya que este juego contiene la generación 1 hasta la 5 de pokémon. Adjunto un enlace con más detalle sobre esto de las generaciones: [Generación Pokémon](#)

Volviendo a la pantalla, en general, el apartado del equipo se irá actualizando. Es decir, si acabas de tener un combate y tu Pokémon Blaziken ha recibido daños o incluso esta a 0 HP, lo podrás consultar aquí.



Pantalla Tus Pokémon (PC)

Como se ha visto con anterioridad esta es una de las opciones del menú principal, esta es la pantalla donde puedes gestionar tus Pokémon. A simple vista verás en el centro un gran panel semitransparente con iconos de todos los Pokémon que has capturado, cada caja corresponde a una caja distinta (aquí estamos en “Caja 1”, que puedes cambiar con las flechas izquierda/derecha en la parte superior), solo podrás cambiar de caja si tienes suficientes pokémon para ocupar al menos un hueco en esa caja. Los pequeños sprites te permiten reconocer rápidamente quiénes están ahí: desde ese adorable Phione hasta ese imponente Dialga, esto nos permite identificarlos.

A la derecha tienes el área de información del Pokémon seleccionado. Cuando haces clic en uno de tus pokémon, su sprite aparece ahí mismo y aparecerán sus estadísticas: HP, ataque, defensa, ataque especial, defensa especial y velocidad. Además verás una barra de vida que te indica la vida actual del pokémon, por si necesita una curación antes de volver a tu equipo.



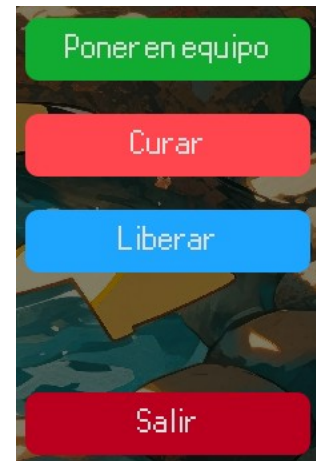
Debajo de esos datos encuentras cuatro botones de acción:

Poner en equipo (verde): mueve al Pokémon de la caja a tu equipo.

Curar (rojo): restaura su HP y elimina estados alterados. Ideal para dejarlo listo antes de lanzarlo a la batalla.

Liberar (azul): elimina al Pokémon de tu colección para siempre, liberando espacio en las cajas. Ten cuidado: no hay marcha atrás.

Salir (rojo oscuro): cierra esta pantalla y regresa al mapa o menú principal sin hacer cambios.



Poner en equipo: mueve al Pokémon de la caja a tu equipo, en la primera ranura libre. Si tu equipo ya está lleno (máximo seis), te avisará y te saldrá una ventana para reemplazarlo por uno de ellos, aquí veremos un ejemplo al respecto, seleccionando incluso un pokémon para ver como se ve:



Moverse entre las cajas y los Pokémon es muy fluido: usa el ratón para seleccionar. Conforme cambias de caja, verás cómo los sprites se actualizan y la etiqueta superior indica “Caja 2”, “Caja 3”, etc. Así puedes ver todos los pokémon que tienes guardados.

En resumen, esta interfaz está pensada para que gestionas tu “inventario pokémon” de forma rápida y visual, sin complicaciones. Donde decides quién entra en tu equipo, quién descansa en el PC.

Pantallas secundarias

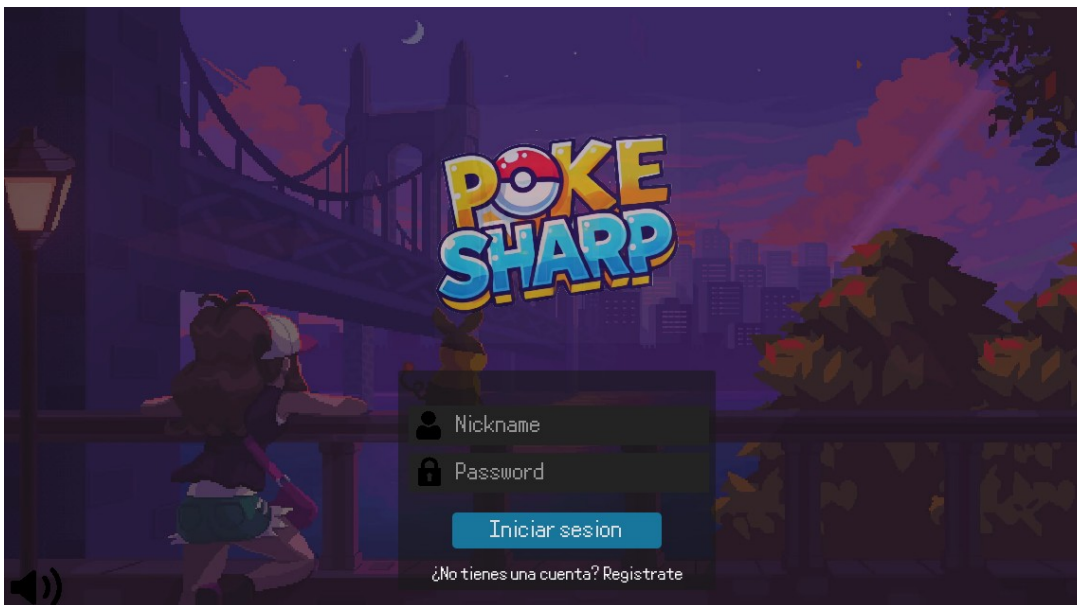
Estas pantallas son secundarias debido a su impacto al juego, carecen de contenido avanzado técnico y no suponen algo demasiado complejo de realizar.

Inicio de sesión

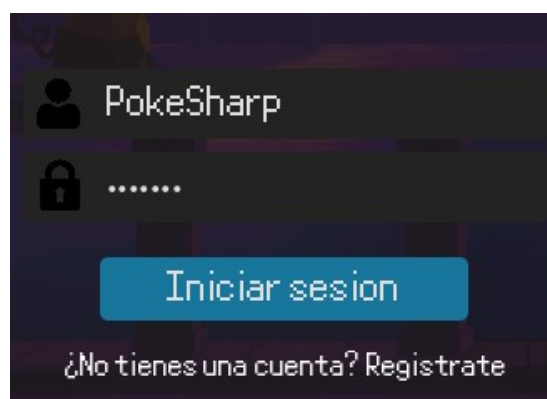
Esta pantalla se muestra al ejecutar el juego, siempre, sin excepción, es una forma de ver que jugador se encuentra jugando para poder consultar a la base de datos los datos requeridos, después iremos paso a paso de como se gestiona todo esto por la parte de backend.

Primero veamos la interfaz, empieza a sonar la música, el logo de PokeSharp hace una animación de tipo latido, volviéndose más grande y pequeño como si se tratará de un corazón latiendo, el fondo es animado, y a la parte de abajo a la izquierda vemos un altavoz para silenciar la música del inicio de sesión, esta simplemente la silencia de inmediato.

La parte más importante es la parte central de la parte inferior, donde tenemos una caja que contiene para introducir el usuario y la contraseña



Si introducimos la contraseña nos aparecerá de la siguiente forma, para hacer más segura la interfaz:



Vamos a probar a registrarnos, ponemos un usuario y una contraseña, ¡Pero cuidado! Este usuario debe ser único, es decir, no pueden existir dos iguales, veamos lo que ocurre si ya existe uno, nos salta un aviso de que el nickname, el usuario, ya está creado en la base de datos, es por eso que no nos deja, al probar con otro que no exista nos dejará, le damos a “Ya tienes una cuenta? Inicia sesión” y cambiaremos el modo para poder iniciar sesión.



Le damos a “Iniciar sesión” y en caso de que las credenciales coincidan cambiará la parte inferior a lo siguiente, de aquí podemos darle a “¡Empieza tu aventura!” y nos llevará directos al juego, sin ningún tipo de pantalla de carga:



Vamos a ver la parte de backend, al registrarse como hemos visto antes hace alguna comprobación, vamos a ir por partes.

Aquí es el inicio del método que se llama al presionar el botón, esta todo encapsulado con un try catch, primero intentamos obtener el usuario mediante el nickname que ha introducido en el TextEdit de nickname.

```
public async void OnBtnRegistrarsePressedAsync()
{
    try
    {
        // Comprobar el nickname por si es único
        PlayersControllers playersControllers = new PlayersControllers();
        Player player = null;

        player = await PlayersControllers.GetPlayerByNickname(nickname.Text);
    }
}
```

Aquí si es null ya sabemos que no es un usuario que ya ha sido introducido, y si no es null ya que nos está devolviendo un jugador ponemos que este nickname ya está utilizado, como hemos visto anteriormente.

Para el registro lo que hace es serializar un JSON con el nombre de usuario y la contraseña y lo manda.

```
if (player != null) {
    lblError.Text = "¡El nickname utilizado ya existe!";
} else {
    await playersControllers.InsertPlayer(nickname.Text, password.Text);
}
```

Aquí por una parte, otorgamos un pokémon inicial para que tenga algún pokémon, calculamos sus estadísticas, que básicamente en lo que se basa es en coger el nivel y hacer cálculos con las estadísticas base que tiene el pokémon y se lo otorgamos al jugador, asignamos en una variable estática.

```
Pokemon pokemonStarter = new Pokemon();

pokemonStarter = await PokemonController.GetPokemonById(4);
pokemonStarter.CalcularStats();

Game.PlayerPlaying = player;

var pokemonPlayersController = new PokemonPlayersController();
bool added = await pokemonPlayersController.CapturarPokemon(pokemonStarter, GetTree().Root.GetNode("Game"));
```


Ahora vamos a ver el inicio de sesión.

El inicio es similar al registro, hace el login y si encuentra el login en este caso asigna directamente el jugador y luego asigna los pokémon del jugador, que ahora entraremos en detalle sobre eso.

```
PlayersControllers playersControllers = new PlayersControllers();
Player player = null;

player = await PlayersControllers.Login(nickname.Text, password.Text);

if (player != null) {
    Game.PlayerPlaying = player;

    await player.asignarPokemons();
}
```

En asignar pokémon obtenemos todos los pokémon sin excepción del jugador, a partir de ahí a cada pokémon se le vuelven a calcular las estadísticas y dependiendo de la variable inTeam identificamos si se encuentra en el PC o en el equipo, para evitar duplicaciones hacemos un contains en negativo para que no haya pokémon duplicados.

A cada pokémon que entre al equipo se le obtiene un moveset, que es un set de movimientos pokémon, en caso de que no lo tenga creado se le genera introduciendo los movimientos en la base de datos, finalmente se añade el pokémon a la lista del equipo del jugador y nos muestra como hemos visto anteriormente la pantalla para poder seguir con nuestra partida.

```
public async Task asignarPokemons()
{
    var pokemonPlayersController = new PokemonPlayersController();
    var listaAllPokemons = await pokemonPlayersController.GetPokemonsByPlayer(id);

    foreach (Pokemon poke in listaAllPokemons) {
        poke.CalcularStats();

        if (poke.inTeam == -1) {
            if (!listPokemonsCaja.Contains(poke)) {
                listPokemonsCaja.Add(poke);
            }
        } else {
            if (!listPokemonsTeam.Contains(poke)) {
                await poke.getMovesetDB();

                if (poke.Movimientos == null || poke.Movimientos.Count < 4) {
                    GD.Print($"Pokémon {poke.NombreCamelCase} con menos de 4 moves");

                    await poke.generateMoveset(true);
                }

                GD.Print($"Pokémon en team:\n {poke}");
                listPokemonsTeam.Add(poke);
            }
        }
    }
}
```

Menú pausa

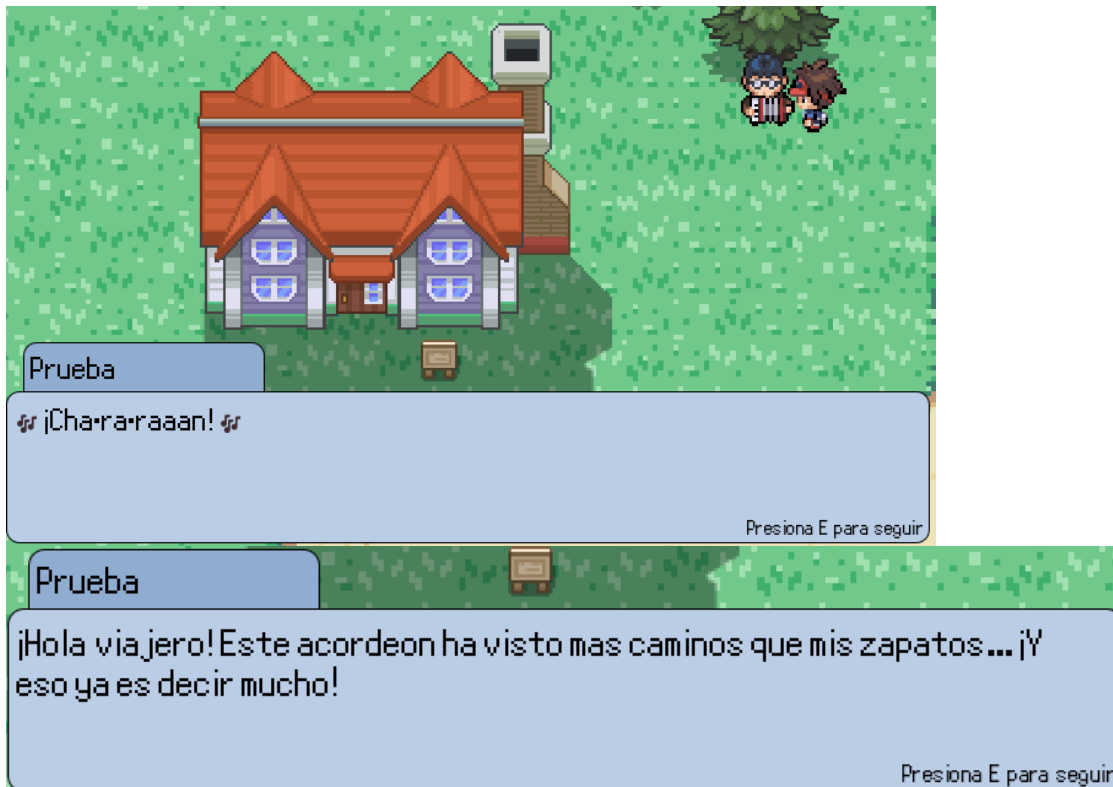
Este menú pausa se accede mediante la tecla “ESC” y también se quita, actualmente no tiene muchas funcionalidades, únicamente el pausar el jugador y también poder volver a la pantalla de inicio.



NPCs

Los NPCs (Non-Player Character) del juego son escasos ya que su sistema actual solo permite el dialogo sin poder conversar tu con el NPC, veamos un ejemplo:

Este NPC tiene una animación mientras se encuentra quieto, cuando nos acercamos y le damos a la “E”, es decir, la tecla para poder interactuar nos salta esta ventana que nos muestra su nombre, que en este caso le puse “Prueba” y abajo el texto que nos dice, y abajo derecha nos avisa de que podemos seguir adelante con el dialogo



Entrenadores

(Pendiente, es lo que falta)

Página web

(Pendiente, al terminar el desarrollo se podrá rellenar)

Control de versiones

(Pendiente, al terminar el desarrollo se podrá)

Discord

¿Qué es Discord?

Discord es algo muy conocido en el mundo de los videojuegos, es una plataforma digital que permite unir a comunidades y también poder comunicarte con otros usuarios, aquí dejo con más detalle al respecto: [Discord: qué es y para qué sirve](#)

¿Por qué hice un servidor de Discord?

Normalmente los juegos suelen tener un servidor de Discord oficial, es la mejor forma de notificar a los usuarios de nuevas actualizaciones, avances o directamente recibir comentarios sobre que les parece el juego, me parecía una gran oportunidad el hacer uno y decidí crear uno, para mantener a una comunidad unida si existe en un futuro.

También porque puede llegar a ser una forma de publicitar el juego, entre amigos en vez de mandar la página web o el repositorio de GitHub puedes mandar una invitación al servidor y de ahí que le interese o no el proyecto.

¿Realmente es necesario?

Si y no, al fin al cabo el unir a una comunidad cuesta, y que se mantengan unidos es más difícil, por eso existen cosas como [Reddit](#), que podría llegar a decirse que es parecido a Discord pero en este caso Reddit es más una especie de foro.

El compartir tu experiencia jugando a un juego solitario con otros puede llegar a derivar en más entretenimiento para todos.

¿Cuál es el uso que le doy para el proyecto?

Como ya dije con anterioridad principalmente es para mantener a una comunidad unida, pero también es para ir anunciando las nuevas versiones del proyecto, posibles avances, recibir reportes de bugs encontrados y sugerencias de las jugadores.

Durante su desarrollo cuando ya era un juego más o menos con posibilidad de jugarlo publiqué dos versiones alfa, que son versiones muy tempranas de su desarrollo y hago una recopilación de lo que está incorporado en la versión, aunque esto también se encuentra en GitHub.

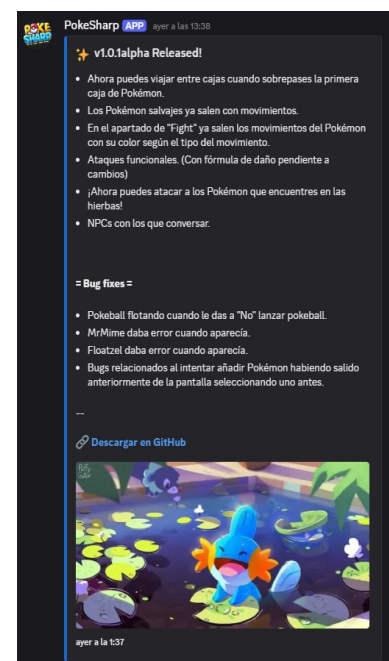


Diagrama Entidad-Relación

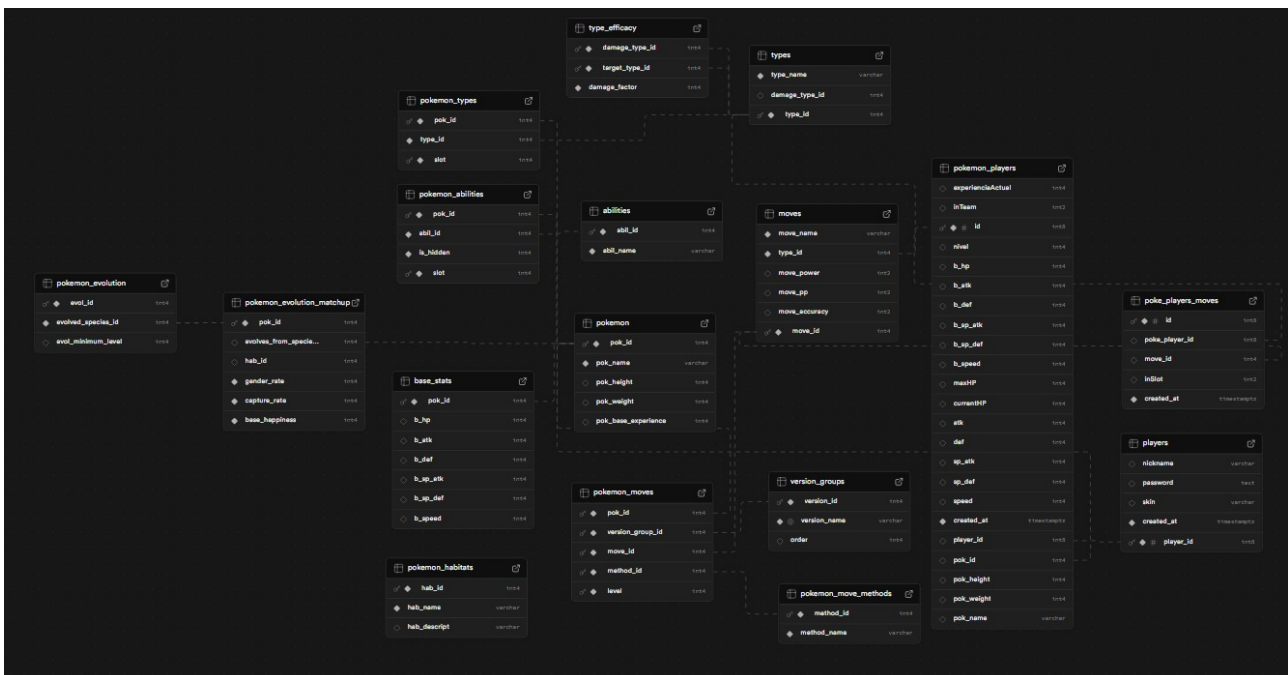
La base de datos utilizada en el proyecto está compuesta por un total de **17 tablas**. De estas, **14** provienen de una base de datos pública relacionada con Pokémon, lo que permitió ahorrar tiempo en la estructuración inicial y centrarme en la lógica del juego.

Las **3 tablas restantes** han sido diseñadas específicamente para gestionar la información de los jugadores. Gracias a ellas, es posible relacionar a cada usuario con sus respectivos Pokémon, así como con otros datos importantes como:

- Sus **credenciales de inicio de sesión**, necesarias para acceder al juego con su cuenta.
- La **asignación de pokémon a cada jugador**, permitiendo así almacenar su progreso.
- Los **movimientos que conoce cada uno de sus pokémon**, lo que permite que cada pokémon tenga habilidades únicas dentro del combate.

Estas tablas personalizadas han sido necesarias para otorgar al proyecto de un sistema de guardado automático, permitiendo un sistema de usuario funcional que se guarda cada vez que ocurre algo importante.

Base de datos de Pokémon: [brianr852/Pokemon-Database](https://github.com/brianr852/Pokemon-Database)



Esquema funcional

Dificultades encontradas y cómo las ha superado

Aprendizaje de nuevas tecnologías

Una de las principales dificultades a lo largo del proyecto ha sido la falta de conocimientos previos en muchas de las tecnologías necesarias. Para desarrollar el juego, fue imprescindible aprender desde cero aspectos como:

- El motor de videojuegos **Godot Engine**
- El lenguaje de programación **C#**, de forma más profunda
- El uso de **Supabase**, una plataforma que utiliza **PostgreSQL**

Superar esta dificultad supuso dedicar muchas más horas de lo realmente planeado. Principalmente usando el método de prueba y error, junto a mucha investigación. Este enfoque me permitió no solo resolver los problemas concretos que iban surgiendo, sino también adquirir un conocimiento sólido de cada nueva herramienta que incorporaba al proyecto.

Problemas con mayúsculas en los scripts

Al realizar la exportación de la primera versión alfa disponible tuve bastantes problemas que me dejaron atascado unos días en ello, el problema era que al probar el juego exportado no funcionaba nada como de normal, lo que hacía que esto suceda era las mayúsculas en algunos scripts, esto por supuesto se solucionó cambiando a todos los scripts a minúsculas, a raíz de eso todo funcionó.

Ambición frente a limitaciones de tiempo

Desde el inicio, la idea del proyecto era mucho más ambiciosa que el resultado actual. Sin embargo, pronto me di cuenta de que muchas de las funcionalidades que había planteado eran más complejas de lo que parecían, y que el tiempo disponible no iba a ser suficiente para realizarlas todas.

Ante esta situación, decidí reorganizar el desarrollo del juego, dividiendo las tareas en tres niveles de prioridad: sistemas principales, secundarias y complementarias. De esta forma me permitió centrarme en construir una base sólida del juego, asegurando un resultado funcional, sobre el que se podrían seguir construyendo futuras mejoras.

Diseño

Estructura carpetas

Implementación

Ideas a futuro

Experiencia personal y conclusiones finales

La experiencia personal durante este proyecto ha sido muy positiva. De hecho, ha sido tan interesante que me ha llevado a plantearme la posibilidad de especializarme en desarrollo de videojuegos. Volviendo al proyecto en sí, ha sido una experiencia que sin duda repetiría, ya que he aprendido muchísimo tanto a nivel técnico como a nivel personal. Uno de los aspectos más valiosos ha sido mejorar mi capacidad de organización y gestión del tiempo para cumplir con los plazos establecidos.

Gracias a este trabajo, he comprendido que llevar a cabo un proyecto sólido y bien hecho no depende exclusivamente del talento, sino de la dedicación y el tiempo que se le invierte.

La principal conclusión que saco de este proyecto es que ha sido, sin duda, la mejor elección que podría haber hecho. Es un proyecto con el que he disfrutado incluso en los momentos que estuve cansado.

Siento que el resultado cumple con las expectativas que tenía al inicio. A futuro, me gustaría seguir trabajando en este juego como hobby, ampliando sus funcionalidades y puliendo aquellos aspectos que podrían mejorarse. Este proyecto ha sido una excelente oportunidad para aprender, crecer y confirmar mi interés por el desarrollo de videojuegos.

Referencias bibliográficas y webgrafía

Imágenes utilizadas:

- Imagen utilizada en la pantalla de “Tus Pokémon” para desplazarte entre cajas: [Flaticon – flecha](#)
- Fondo de inicio de sesión: [Hilda And Tepig Watching Castelia City Pokemon - MoeWalls](#)

Enlaces consultados:

- Documentación de Godot 4 C#: [C#/.NET — Godot Engine \(latest\) documentation in English](#)

Agradecimientos

- [Brianr852](#), un usuario de GitHub que publicó una base de datos muy completa de Pokémon.
- A todas esas personas que suben texturas de Pokémon por internet.
- A las personas encargadas de [Pokémon Essentials](#) por hacer los fan games de Pokémon posibles, otorgando muchas texturas e información de ayuda.
- Especial agradecimiento a Izan López por concederme el permiso para poder empezar de cero el proyecto que tuvimos pensado hacer en un pasado, y también por haber probado el alfa 1.0.0.
- También agradecer a ciertos amigos por probar el juego en su versión alfa, y a una amiga por hacer retoques al logo del juego.