

1.	INTRODUCCIÓN.....	1
2.	OBJETIVOS Y CAMPO DE APLICACIÓN.....	2
3.	DESCRIPCIÓN DEL TRABAJO.....	3
4.	ESTADO DEL ARTE.....	5
5.	DIAGRAMA DE CASOS DE USO.....	7
6.	PROTOTIPADO.....	9
7.	DISEÑO BBDD.....	17
8.	DESARROLLO DE PERFILES DE USUARIOS / ROLES.....	19
9.	DESARROLLO DE CONCURRENCIA EN LA APLICACIÓN.....	21
10.	DESARROLLO EMPRESARIAL.....	21

*Primera y segunda entrega.***1. INTRODUCCIÓN.**

La aplicación tendrá como objetivo principal la consulta de diferentes informaciones sobre todo el mundo Pokémon disponible en la API **PokéAPI**, además permitirá tener enfrentamientos con equipos de hasta 6 Pokémon, contra la máquina, con una duración por turno y una duración que englobará al enfrentamiento, y presentará estadísticas de los enfrentamientos como si se pierde, se gana o se empata por falta de tiempo para hacer un ranking general con los demás jugadores (usuarios registrados).

Antes de jugar los partidos, los usuarios deberán crear equipos personalizados, pudiendo crear hasta 5 equipos, los cuales podrán manipular en cualquier momento, eliminándolos o creando nuevos en el existente.

Para acceder y manipular datos almacenados en una base de datos remota, la aplicación se conectará una vez iniciado sesión y se llevará su sesión a lo largo de la aplicación, para tener en consideración sus favoritos, etc. La API usada permitirá acceder a datos y crear funcionalidades que permita interactuar de otro modo distinto en donde tenga sentido y lógica para su organización.

La base de datos remota a utilizar será ***db4free***, una base de datos gratuita y accesible en línea que ofrece amplia compatibilidad con una variedad de lenguajes de programación y herramientas de desarrollo. Además, al utilizar una base de datos remota se elimina la necesidad de configurar y mantener una base de datos local, lo que permite trabajar en distintos equipos mediante el uso de la herramienta ***Github*** sin miedo a desconfigurar el acceso a la base de datos.

Para el desarrollo de la aplicación se utilizará el lenguaje de programación C# junto a XAML para las ventanas, el cual ofrece facilidad de uso y gran compatibilidad con la plataforma .NET para usar herramientas en caso de ser necesarias. La herramienta de desarrollo a utilizar será **Microsoft Visual 2022**, que proporciona una amplia gama de herramientas y características para desarrollar aplicaciones modernas de manera eficiente y rápida.

Además de la aplicación para las distintas ventanas, se tratará de realizar una página web, eliminando así el registro desde la aplicación y llevándolo a la página web usando lenguajes

Primera y segunda entrega.

de marca como *HTML* y *CSS* y otros lenguajes compatibles con el mundo web como serían *JavaScript* y algo de *PHP* para conectarse a la base de datos e insertar datos y recogerlos, usando **Visual Studio Code** para estos lenguajes.

En resumen, el proyecto consiste en el desarrollo de una aplicación de escritorio que utilizará una API externa para consultar información sobre el juego/serie "Pokémon" y permitirá jugar enfrentamientos con equipos personalizados. La aplicación manipulará datos almacenados en una base de datos remota, la cual será accesible mediante la aplicación, y será desarrollada utilizando el lenguaje de programación C# y la herramienta de desarrollo **Microsoft Visual 2022**. Además, se implementará una página web en relación con la aplicación para poder descargar un ejecutable, registrarse y algún que otro funcionamiento más.

2. OBJETIVOS Y CAMPO DE APLICACIÓN.

Se espera crear una aplicación para uso personal que pueda ser mostrada a empresas como una muestra de trabajo innovador. Se planea investigar aplicaciones existentes para usar como referencia y crear una solución única y distintiva, mejorándola con ideas en base a esas aplicaciones.

Se busca crear una aplicación eficaz y productiva sin bloqueos y se tendrán en cuenta los procesos a utilizar, como las llamadas a la API, la reducción de los tiempos de carga y la identificación de las partes que más tardan en cargar. La idea es crear una aplicación fácil de usar para mejorar la experiencia de los usuarios y ofrecer un servicio de calidad y sencillo visualmente.

Se aspira a crear una aplicación de amplio uso con varias opciones de contenido en la API y múltiples niveles de usuario y administrador. Las características de la aplicación dependerán del tiempo disponible y del proceso de creación de la página web y distintas funcionalidades a implementar de lo que se espera. Se planea utilizar niveles de usuario y administrador para gestionar el acceso a la aplicación.

Los niveles de Usuario y Administrador que se planean utilizar, con una ínfima explicación son:

➤ **Usuarios:**

Primera y segunda entrega.

- » 0, nivel básico, gratuito. Acceso limitado a visualizar contenido de la API, sólo de los Pokemon con distinta información.
- » 1, nivel básico, pero de pago. Se añadirán más opciones de visualización, donde se podrá visualizar, además de lo anterior, distintos niveles de apartados, como sería una pokedex.
- » 2, nivel medio, pago. Creación de equipos y acceso a los contenidos anteriores sumando nuevos apartados.
- » 3, nivel alto, pago. Jugar con el equipo que pueda crear, con las estadísticas y acceso total a los nuevos apartados.

➤ **Administradores:**

- » 1, nivel básico de administrador, podrá acceder a todo el acceso de administrador pero con limitaciones de modificación sobre usuarios.
- » 2, nivel alto de administración, sería un super administrador. Podrá cambiar a los administradores y tener un control total sobre los usuarios.

Se usará paletas de colores y estilos de fuentes similares a las de la serie y/o juego para que sea más inmersivo.

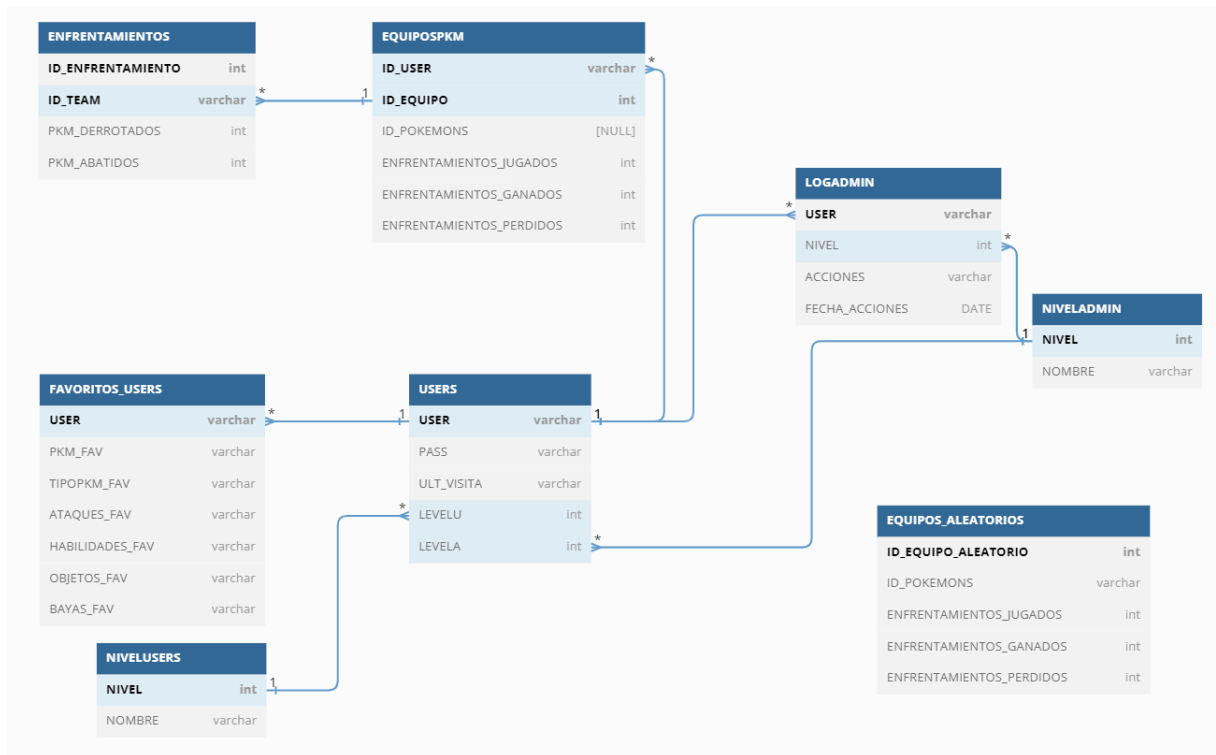
Según se vaya realizando el trabajo, se irán haciendo más opciones, ya que surgirán más ideas en su desarrollo, de forma espontánea que serán explicadas en todo momento.

3. DESCRIPCIÓN DEL TRABAJO.

Se utilizará el siguiente esquema de base de datos: [podrá ir variando según se vaya avanzando la aplicación, pero será parecido al esquema]

[DIAGRAMA DESACTUALIZADO, EN EL APARTADO DE BBDD ESTÁ ACTUALIZADO]

Primera y segunda entrega.



Como ya se comentó anteriormente, en el primer apartado de este documento, la aplicación que se desarrollará tiene como objetivo el de consultar de distintos tipos de información que se considere interesante para ir añadiendo contenido a la aplicación.

Otra opción será la creación de equipos, será una opción que permitirá crear un máximo de 5 equipos y tenerlos almacenarlos al nombre del usuario. Se podrán elegir todos los Pokémon que haya disponibles. Estos equipos se podrán manipular en cualquier momento, eliminarlos o crearlos de nuevo en el existente.

Una vez creados equipos, estará disponible la opción de jugar enfrentamientos, donde se crearán equipos aleatorios que estarán en la tabla "EQUIPOS_ALEATORIOS" a los cuales enfrentarte, se hará como un hilo, con una duración de X minutos que vaya cronometrando cuando pase el tiempo y, una vez finalizado dicho enfrentamiento, se almacenará en BBDD el resultado y se pondrá en la tabla equipos el enfrentamiento jugado, perdido o ganado.

Finalizado el enfrentamiento, se podrá acceder a un tipo de ranking donde saldrá todos los resultados de los equipos creados globalmente en la base de datos, ordenado el *select* por partidos ganados y partidos jugados, haciendo una media de ganados. De esta manera, saldrá

Primera y segunda entrega.

el primero en la clasificación y se irán mostrando. Ese Ranking general, se hará de los demás jugadores (usuarios registrados).

Las opciones del administrador se almacenarán en una tabla de registro, dónde se creará un nuevo registro por cada acción que realice el administrador [tabla LOGADMIN]. Al tener dos niveles distintos de administración, el Super Administrador tendrá acceso a otro tipo de opciones, además de las de Administrador y no se irá registrando las opciones del Super Administrador.

4. ESTADO DEL ARTE.

Respecto a la aplicación, para poder hacerla correctamente se ha de tener unos conocimientos mínimos sobre Pokémon y saber las mecánicas que se usan dentro de los videojuegos para poder llevarlas a la aplicación lo más parecido posible para que a la parte Usuario les parezca sencillo de usar y sea más familiar. Para ello, primeramente, se ha tenido que llevar una investigación sobre diferentes videojuegos a lo largo de la historia para ver cómo ha ido evolucionando y como se podría llevar a cabo la implementación. [1, 2, 3, 4, 5] (Cuando haya que poner referencias, se pondrán en los números el enlace a esa referencia al final del documento, por el momento no lo he puesto)

Para la idea de los combates entre equipos distintos me he orientado en el programa en línea de **Pokémon Showdown**, una famosa página que representa muy bien lo que es un combate Pokémon. En ella hay diferentes generaciones, diferentes salas en las que te puedes comunicar con las personas, como si fuera una red social, ver batallas en directo de otras personas, ver la lista de ranking o incluso participar en torneos que se lleven a cabo. Hay distintos formatos, bastante variados como se ve en la imagen.

Primera y segunda entrega.

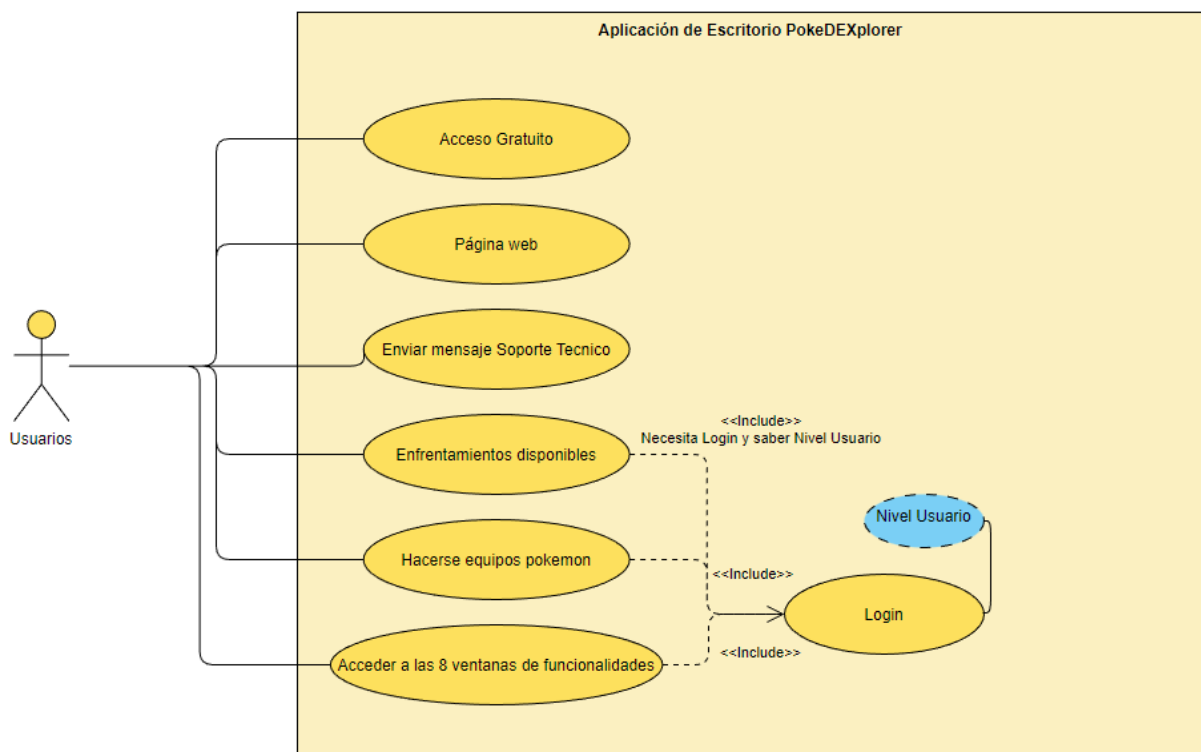
Otras aplicaciones / juegos que se han estado viendo a lo largo del desarrollo son los juegos más novedosos de la compañía, además de juegos hechos por pequeños desarrolladores como adaptaciones a los juegos ya existentes.

En la parte que sólo muestra información general, es decir, donde se hace la llamada a la API que se está utilizando, se han elegido los apartados más curiosos o los que más uso le das a lo largo de sus videojuegos y/o más se ha visto en la serie pero nunca, los usuarios, se han puesto a investigar datos más técnicos como el daño que hace un ataque, o la tabla de tipos de los distintos Pokémon que existen para saber sus debilidades o las fortalezas y menos aun cuando en las entregas de los juegos más novedosas ya se dice automáticamente para que los jugadores no se tengan que esforzar en buscarlo. También, para la gente que es más curiosa, le podría interesar interesante tener todo en un mismo lugar, sin tener que buscar en diferentes páginas ya que, por ejemplo, una poción que te recupera la vida tiene tres formatos

Primera y segunda entrega.

diferentes, una que cura X, otra que cura Z y otra que cura Y, serían tres páginas diferentes, pero en la aplicación estarían en el mismo lugar, siendo esta forma más rápido.

En referencia al concepto que se está utilizando hay demasiada información, tanto de gente que trabaja en la compañía Nintendo / GameFreak, como gente que hacer adaptaciones, fans de los juegos y del anime y es fácil encontrar cualquier tipo de información o buscar algo de inspiración, por lo que es más llevadero el proyecto y se avanza con algo más de facilidad.

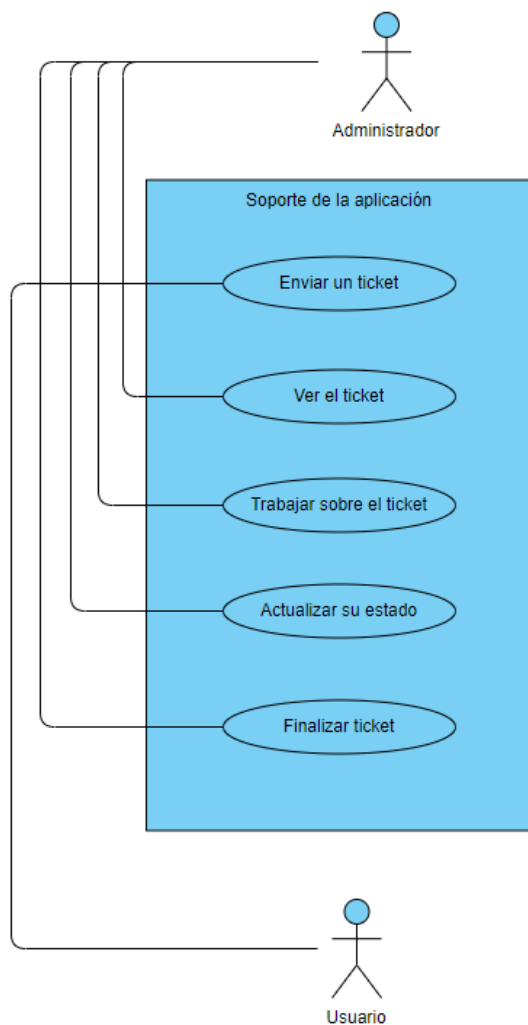
5. DIAGRAMA DE CASOS DE USO.

En este primer Diagrama de casos de uso se explica lo que puede hacer un Usuario de abajo a arriba:

- » Un usuario puede acceder a las distintas ventanas de funcionalidades que están disponibles en la aplicación [8 en total según nivel], podrá crear un máximo de 5 equipos Pokémon y tendrá, si su nivel lo permite, acceso a los enfrentamientos contra la máquina. Para acceder a esto, necesitará iniciar sesión y saber su nivel de usuario, de ahí que tenga la relación con "Login" y sea necesario.

Primera y segunda entrega.

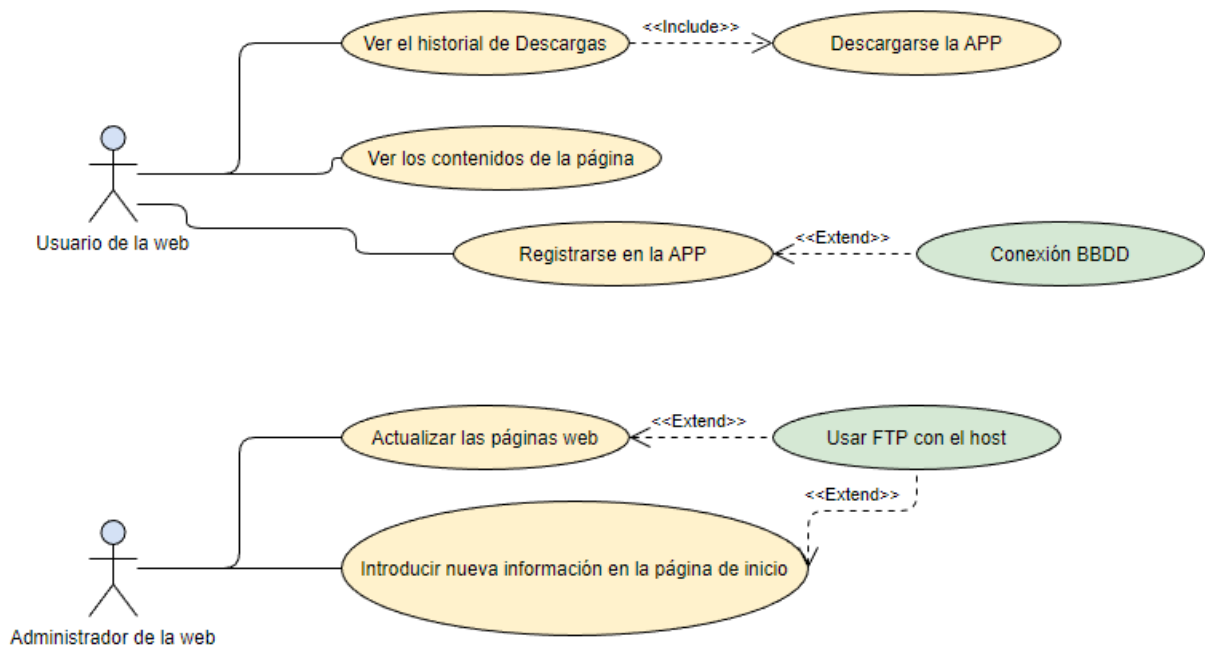
- » La segunda parte del Diagrama hace referencia a los Usuarios que no inician sesión, como puede ser acceso a la parte gratuita del programa, el acceso a la página web de la aplicación y enviar mensajes a Soporte Técnico.



En este diagrama se explica el cómo se va a tratar el tema de soporte técnico / resolución de bugs / explicación de bugs y demás en relación con la aplicación y los usuarios que la utilicen.

- » Un administrador, independiente de su nivel [Nivel 1 o Nivel 2] podrá ver el ticket enviado por el usuario, puede trabajar sobre ese mismo ticket para arreglarlo y, una vez arreglado y/o finalizado podrá actualizar su estado y dar por finalizado el ticket, almacenándose en conjunto con los tickets con estado finalizado.
- » El usuario sólo los podrá crear, se les avisará de que fue enviado correctamente en caso de que sea así y, para saber su estado, pueden ir al apartado de la web de descargas para ver si en la nueva versión se arregló lo que notificaron.

En el siguiente diagrama se explicará el funcionamiento de la página web tanto de la parte del Web Master (administrador) y la parte que el usuario ve.

Primera y segunda entrega.

- » En primer lugar, el usuario de la web puede acceder totalmente gratuitamente a los contenidos que la página pone a disposición de los visitantes, podrá ver las tres páginas que están disponibles actualmente y registrarse para usar posteriormente la Aplicación de escritorio.
- » En segundo lugar, el administrador de la web puede actualizar las distintas páginas web que están disponibles para el usuario mediante el servicio FTP que ofrece el host de la página. Además, podrá añadir información, elementos nuevos y demás a la página cuando sea necesario.

6. PROTOTIPADO.

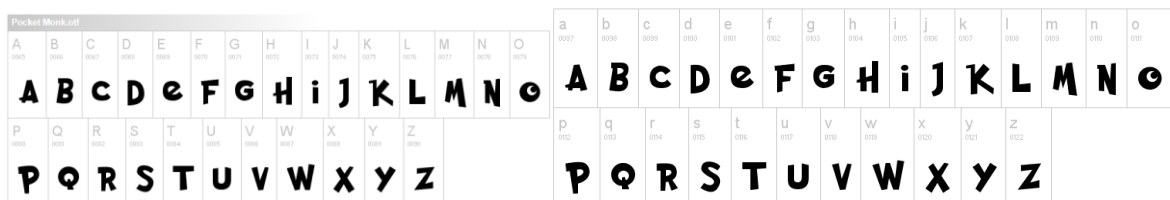
La idea principal e inicial del proyecto a realizar es utilizar una paleta de colores en relación a Pokémon, guiándose por la siguiente página: [paleta de colores](#). Para colores de fondo de pantalla o fondos de algunos elementos de WPF se ha utilizado dos, primero una variación más oscura del color en HTML "SteelBlue", su código en Hexadecimal sería: "#3761a8" y una variación más clara del color ya mencionado, en HTML "LightSteelBlue". Para los colores de letras se ha utilizado el color en HTML "Gold" y "Black" principalmente, utilizando algún otro color para alguna cosa puntual.

Primera y segunda entrega.

Para los estilos de fuentes se han implementado cuatro tipos distintos de letras estilo Pokémon, las cuales son:

Nota: Si se da click al nombre de la fuente se redirigirá a la ventana de descarga de la fuente.

- » **Pocket Monk by Chequered Ink** → Esta ha sido utilizada para, en regla general, títulos y subtítulos y en su mayoría en color “Gold”. Es una letra grande, con una tipografía muy similar a la de Pokémon, dando una inmersión total a la aplicación si se usa. La única desventaja que se ha notado con este tipo de letra es la ínfima diferencia entre letras, como se ve en la siguiente imagen.



- » **Pokemon GB by Jackster Productions** → Este tipo de letra se ha utilizado alrededor de las ventanas de función que hay disponibles, en todas ellas se ha aplicado este estilo. Al igual que pasaba con el anterior, este tipo de letra se utilizó para los juegos más antiguos, es decir, de los primeros que salieron a mercado y, para gente que los jugó, utilizar esta letra puede ser un recuerdo bonito si han jugado a esos juegos, de ahí su uso en la aplicación al ser tan familiar para ese núcleo de personas.



- » **Pokemon Solid by IPBP** → Por el momento no se ha llevado a cabo la implementación de esta fuente. Se usará para la creación de equipos, mezclando el color “Gold” con el color “Black”. Esta letra es menos común, pero tiene también un estilo un poco más moderno sobre la temática Pokémon. La diferencia entre esta letra y la siguiente letra es que esta es “Solid”, es de color negro de relleno y la siguiente es transparente, sólo tiene el color del contorno de la letra.

Primera y segunda entrega.

- » **Pokemon Hollow by IPBP** → Por el momento no se ha llevado a cabo la implementación de esta fuente. Se usará para mostrar el ranking/estadísticas de los combates que se han llevado a cabo.

Los títulos de texto con contorno en color “#3761a8” han sido hechos a mano debido a la limitación de poder poner un contorno a la letra mediante código. Para su creación se ha utilizado tres páginas diferentes, cada una con una función distinta.

- » El primer paso es acceder a esta **página**. En ella se lleva a cabo la cadena de texto que queremos mostrar como título, se elige el color [2] (que se ha de poner manualmente, hasta acertar con el color de la paleta de colores), se elige el tamaño [1] (es más fácil que la elección de color ya que por código sí que se puede aumentar o disminuir su resolución en pantalla) y, por último, se descarga para pasar al siguiente paso [3].



- » El segundo paso es acceder a esta **página**. Como se verá de primera impresión es muy similar a la famosa aplicación de escritorio “Photoshop” pero con la diferencia de que esta es de uso gratuito, perfecto para el uso de la aplicación. Para empezar, se dará un click a la lupa, poniendo en el cuadro de búsqueda “Contorno” [1]. Después, se elegirá los píxeles necesarios, en este caso se ha utilizado 7 en todo momento [2], al igual que el color de contorno [3]. Como se ve, a la hora de elección de color, se puede utilizar diferentes formas, lo cuál puede ser útil en diferentes ámbitos. Por último, se dará a “Sí” para poder confirmar en ambas ventanas emergente y se pondrá el contorno con el estilo seleccionado. [4]

Primera y segunda entrega.



Finalmente, se dará a “Archivo”, presente en la barra de tareas y se descargará un archivo con extensión “psd”

- » La última página que se utilizará es un **convertidor**. Este convertidor nos permitirá convertir la imagen hecha previamente con formato “psd” a formato “png” (hay disponibles más formatos, pero el utilizado en la aplicación es png para todas las imágenes). Es tan fácil como darle a “Convertir” y comenzará la conversión de formato. Una vez terminada dicha conversión, se podrá volver a descargar con el formato que se deseaba en primera instancia.



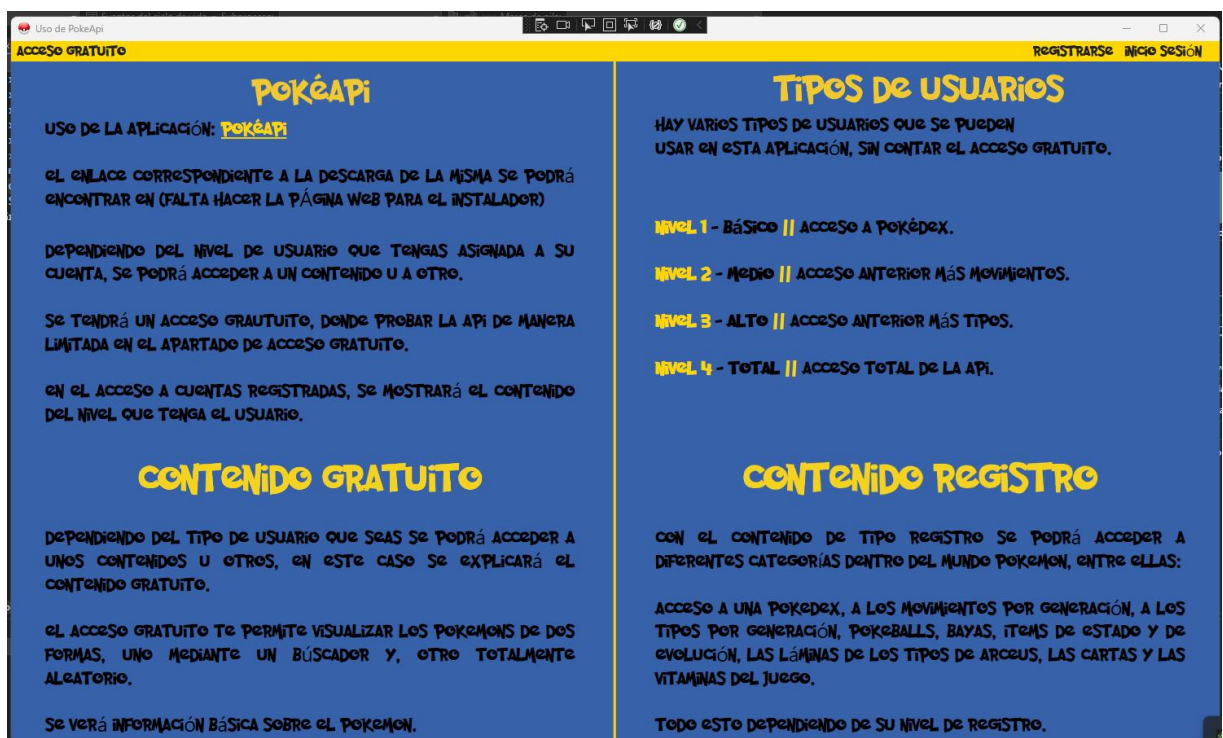
Primera y segunda entrega.

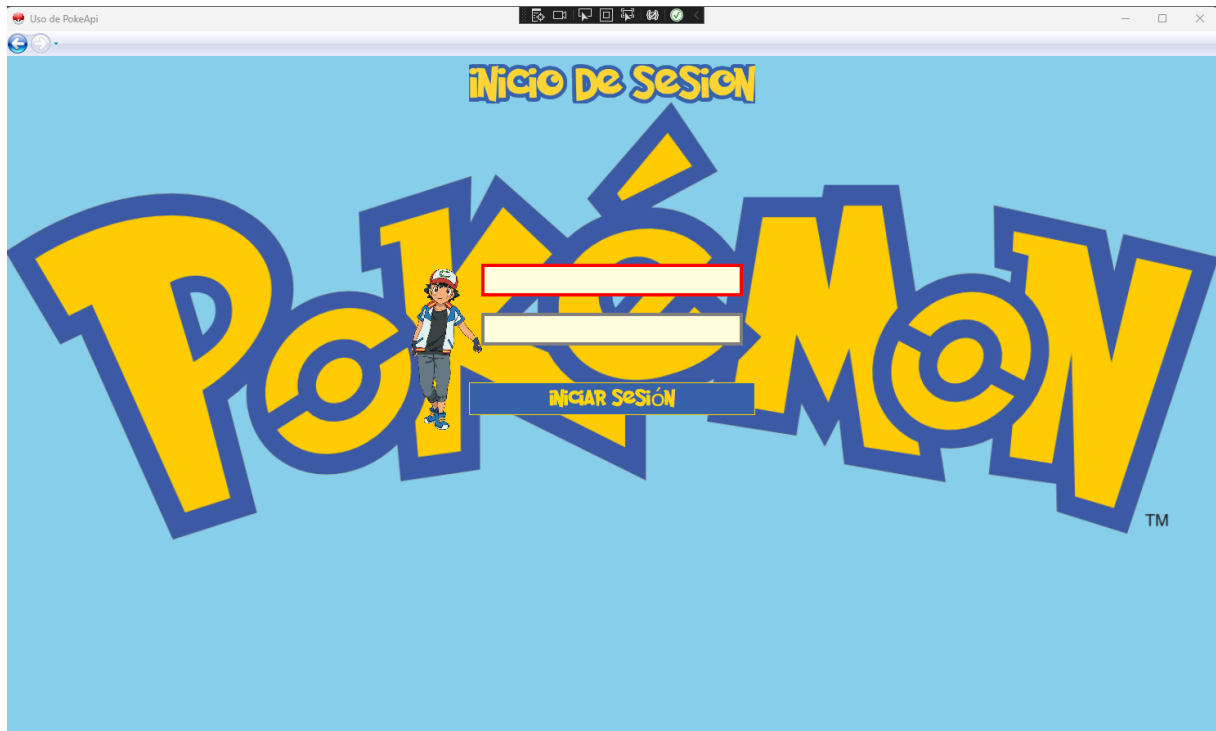
El estilo de ventanas de la aplicación se hará de tipo Navegables, limitando algunas entradas como sería la de volver hacia atrás una vez has iniciado sesión sin primero cerrar sesión, solucionando así que puedas acceder al contenido sin estar iniciado sesión.

Gracias a que sea de estilo Navegable es mucho más fácil al usuario saber donde está y a donde quiere ir, es decir, visualmente es más cómodo que estar yendo a una ventana inicial con las opciones disponibles si ya está presenta en una barra superior, mucho más a la vista y sin la necesidad de aprenderse en donde está el apartado al que quieras ir a través de más páginas.

El tamaño por defecto del proyecto que se está llevando a cabo será de ancho 1500px y de alto 900px, haciendo así que sea más fácil de ver la letra al ser una ventana más grande. Por el momento no tiene resoluciones distintas a las mencionadas, pero es uno de los objetivos, hacer que la aplicación se ajuste a todo tipo de pantallas y que se pueda usar en todo tipo de interfaces visuales y resoluciones.

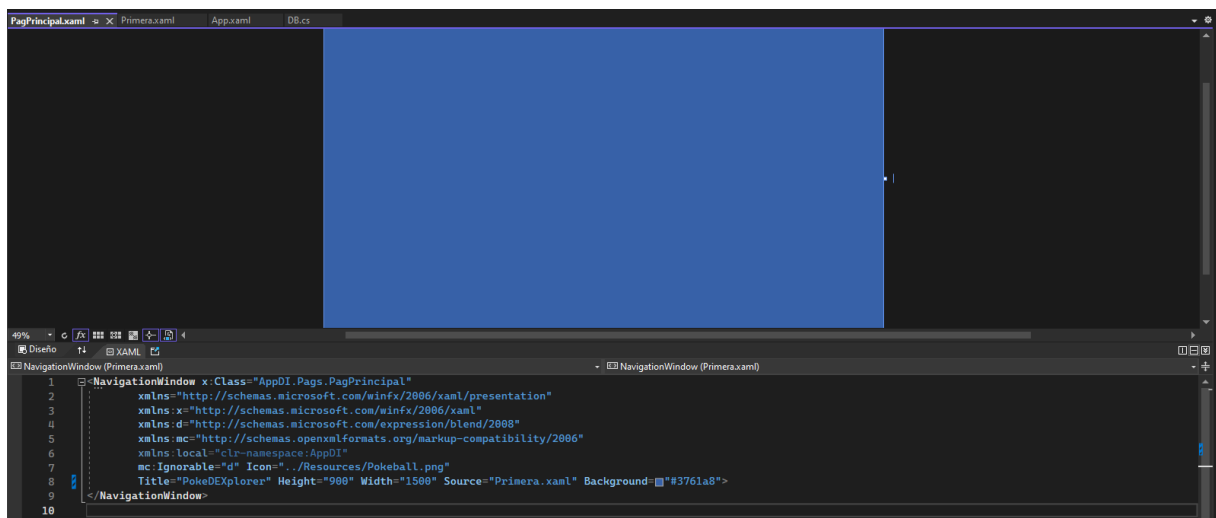
Algunos ejemplos de ventanas de la aplicación para ver el diseño:



Primera y segunda entrega.

Al estar en desarrollo, estas ventanas pueden ser modificadas a lo largo del proyecto según necesidades y según estilos aplicados para hacerla similar en todas las ventanas y se siga un patrón.

Para hacer funcionar el estilo de ventanas navegables se utiliza una ventana que es utilizada como contenedor principal de las demás ventanas, es decir, es como un padre donde sus hijos van yendo según lo asignado.



Primera y segunda entrega.

En relación a los estilos comentando previamente, se usará una plantilla donde se irán almacenando todos los estilos que se vayan aplicando a los elementos WPF de la aplicación, para poder hacer así un código reutilizable y que se uso sea más rápido al no tener que estar aplicando lo mismo para muchos elementos. La ventana es llamada App.xaml, en donde se puede apreciar que cada estilo hecho está nombrado con un comentario para que en el momento donde se tenga bastantes líneas se pueda saber que es cada cosa y que su búsqueda sea rápida.

```
<Application x:Class="AppDI.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:noe="clr-namespace:NoesisGUIExtensions;assembly=Noesis.GUI.Extensions"
    xmlns:local="clr-namespace:AppDI"
    StartupUri="/Pages/PagPrincipal.xaml">
    <Application.Resources>

        <FontFamily x:Key="PkmFont">pack://application:,,,/Fuentes/#Pokemon Hollow</FontFamily>
        <FontFamily x:Key="PkmFontBold">pack://application:,,,/Fuentes/#Pokemon Solid</FontFamily>
        <FontFamily x:Key="PkmMonk">pack://application:,,,/Fuentes/#Pocket Monk</FontFamily>
        <FontFamily x:Key="PkmGB">pack://application:,,,/Fuentes/#Pokemon GB</FontFamily>

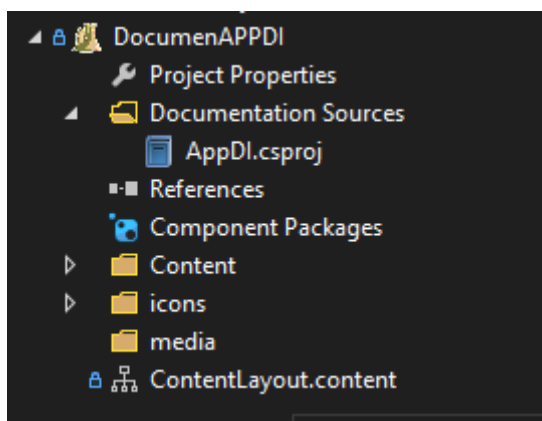
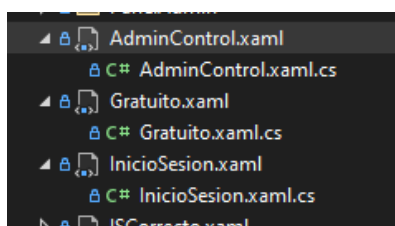
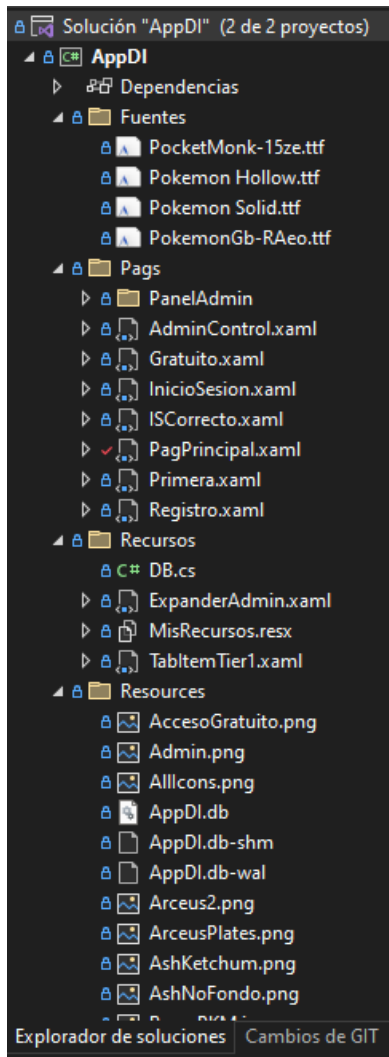
        <!-- Centrar total de un Label. -->
        <Style x:Key="CentradoTotalLb" TargetType="Label">
            <Setter Property="VerticalAlignment" Value="Center"/>
            <Setter Property="HorizontalAlignment" Value="Center"/>
        </Style>

        <!-- Centrar total de un TextBlock -->
        <Style x:Key="CentradoTotalTBlock" TargetType="TextBlock">
            <Setter Property="VerticalAlignment" Value="Center"/>
            <Setter Property="HorizontalAlignment" Value="Center"/>
            <Setter Property="TextAlignment" Value="Center"/>
        </Style>

        <!-- Texto justificado con 25px de tamaño de letra y centrado -->
        <Style x:Key="TxtJustificado25px" TargetType="TextBlock">
            <Setter Property="FontSize" Value="25px"/>
            <Setter Property="TextAlignment" Value="Justify"/>
            <Setter Property="TextWrapping" Value="Wrap"/>
            <Setter Property="HorizontalAlignment" Value="Center"/>
            <Setter Property="VerticalAlignment" Value="Center"/>
        </Style>

        <!-- Para los títulos que son imagen. -->
        <Style x:Key="ImgTitulo" TargetType="Image">
            <Setter Property="Width" Value="350"/>
            <Setter Property="Height" Value="50"/>
            <Setter Property="VerticalAlignment" Value="Top"/>
            <Setter Property="Margin" Value="10"/>
        </Style>
    </Application.Resources>
</Application>
```

El modelo de carpetas tal y como está actualmente está dividido de la siguiente forma:

Primera y segunda entrega.

En la imagen presente en la parte izquierda del documento se ve como está yendo el proyecto, está organizado de tal forma que se tiene los elementos de más código en la parte de “Recursos”, ya que como tal no tienen relación con las páginas, pero hacen la interacción entre la aplicación y lo que el usuario dice, con la parte servidor que utiliza para conectarse a la base de datos hosteada. Es la llamada parte “**Modelo**” ya que está la lógica y hacer que todo sea como un reloj.

Dentro de las ventanas está el apartado general en el cuál es tienen las ventanas de navegación y, dentro de las ventanas, un apartado referente al panel de administrador, las ventanas que utilizará el administrador para llevar a cabo sus funciones. En este mismo apartado, los propios archivos de ventanas tienen su código, que hacen de **controlador** y hacen la comunicación entre esa ventana que maneja el usuario (**la vista**) y, gracias al código detrás de la vista, se lleva a la carpeta “Recursos” para que esos elementos dentro hagan de **Modelo** y que funcione todo.

En la carpeta de Fuentes se tienen todos los tipos utilizados de fuentes en la aplicación.

Por último, la carpeta de recursos que se utilizará para almacenar los recursos insertado en el archivo de “Recursos” → “MisRecursos.resx”. Es una carpeta que no se puede modificar ya que puede llevar a errores.

Otra parte de lo que se está realizando es la parte de documentación. En ella vendrá explicado los métodos realizados y demás información útil dentro del código para que se entienda mejor y sea más fácil saber su funcionamiento y su idea del porqué.

Primera y segunda entrega.

7. DISEÑO BBDD.

Diagrama con relaciones y campos de la Base de Datos: [Página web utilizada]

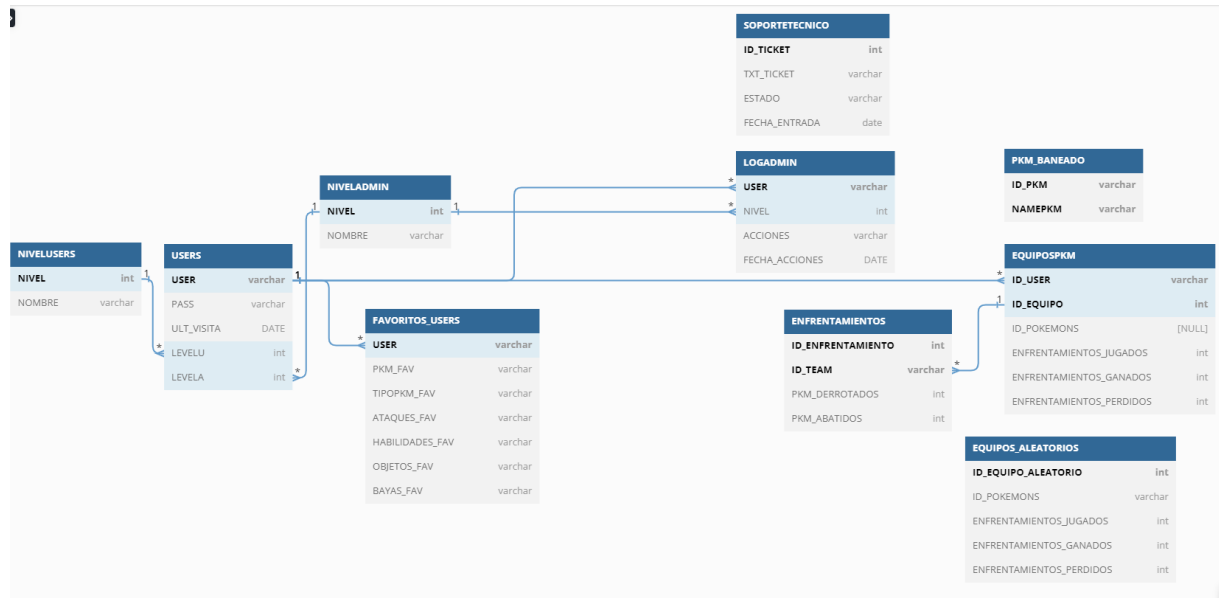
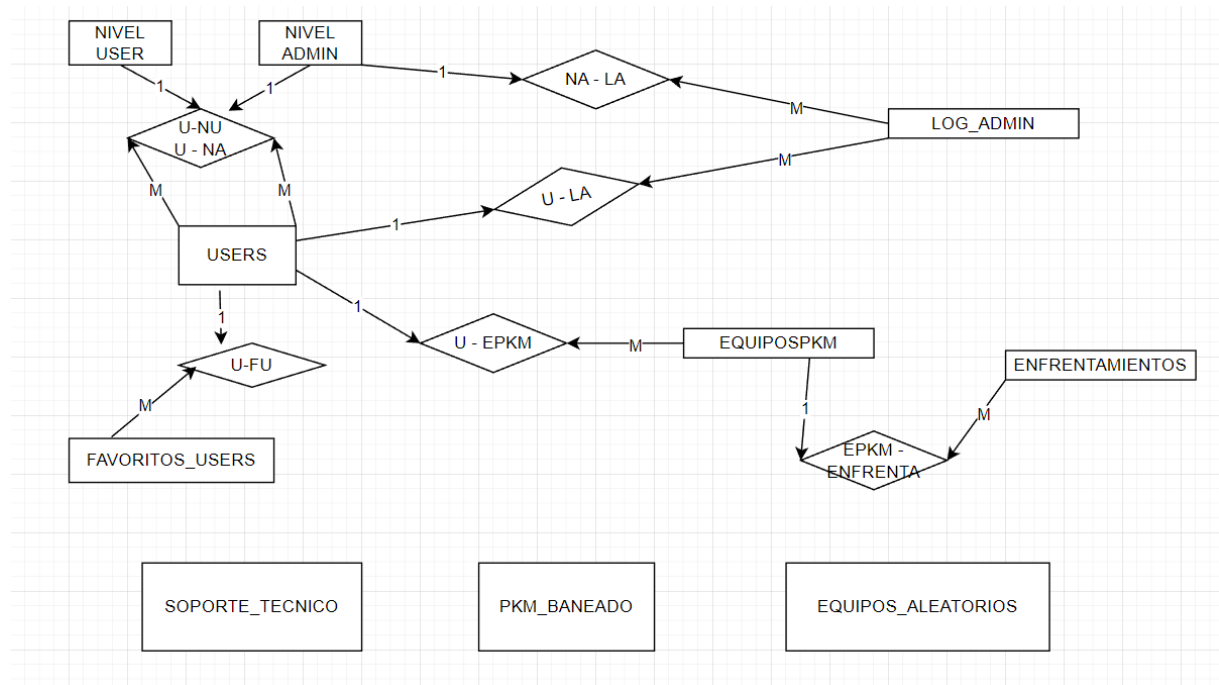


Diagrama conceptual de la base de datos:



En total, por el momento, se tiene 10 tablas, cada una para un propósito diferente.

En primera instancia se explicará el funcionamiento de la tabla denominada "USERS":

Primera y segunda entrega.

- » USERS → Tabla que se utiliza para el control de usuarios, tiene una relación de muchos a uno con los niveles, para saber qué tipo de niveles tienen de administrador y de usuario. Para registrar campos en esta tabla se utilizará la página web, donde se hará un registro para posteriormente hashear la contraseña con un código salt. Una vez que un usuario se ha registrado se creará el valor, por defecto será nivel 1 de usuario y nivel 0 de administrador. Cuando se inicie sesión, se ejecutará un comando de *update* para el campo de tipo Date de última visita. Como se ve tiene varias relaciones con otras tablas “LOGADMIN”, “FAVORITOS_USERS” y “EQUIPOSKMN”.
- › LOGADMIN → Tabla utilizada para saber que hacen y cuando hacen cambios sobre las tablas los administradores, se sabrá que hacen y cuando lo hacen, con un nombre de quien lo ha hecho.
- › FAVORITOS_USERS → En esta tabla se registrará cuando un usuario le de a favoritos o quite el favorito, borrando el registro. Se podrá tener una ventana en la cual se muestren todos los favoritos que este usuario tiene, los datos se almacenan como tipo JSON, por campos clave valor para tener la información más condensada y fácil de representar a posterior.
- › EQUIPOSPKM → Esta tabla se irá actualizando cuando un usuario que tenga el nivel correspondiente cree un equipo, en ella se almacenará el ID que tenga el equipo que se autoincrementará y los Pokémon que corresponden a ese equipo creado. A su vez, esta tabla tiene otra relación con la tabla Enfrentamientos:
 - ENFRENTAMIENTOS → Se usará para saber con que equipo se ha peleado, a los Pokémon que ha derrotado en general y a los Pokémon que le han matado en general.
- » TABLAS INDEPENDIENTES → Estas tablas como se ve en ambos diagramas insertados anteriormente, no tienen ningún tipo de relación con nada ya que son usadas meramente para dar apoyo o para usarse para algún motivo que no tiene relación con las demás tablas creadas y no afecta al funcionamiento de la aplicación, sino que dan apoyo para que haya más contenido y sea una aplicación con más opciones.

Primera y segunda entrega.

- › SOPORTE_TECNICO → Esta tabla no tiene relación a un usuario registrado, aunque sea el usuario registrado el que haga el ticket en primer lugar. La explicación es que es indiferente quien reporte los bugs, ya que no se le daría las gracias, únicamente porta información que se necesite el texto, que en este caso sería el bug/sugerencia y que verá el administrador para dar solución. Los errores que se soluciones irán a la página web, en el apartado de Descargas conjuntamente a una nueva actualización.
- › PKM_BANEADO → Es una tabla que únicamente tendrá acceso el grupo de administradores, se usará para seleccionar Pokémon y banearlos para que los usuarios no lo utilicen a la hora de crear sus equipos. Su funcionamiento planteado es para cuando el usuario introduzca el nombre del Pokémon, se realice una selección de todos los datos de esta tabla y, si el ID de ese Pokémon seleccionado está en la tabla no permitirá su selección y por ello no necesita relación con nada, sólo se buscará información y se introducirá o retirará.
- › EQUIPOS_ALEATORIOS → Serán equipos que se creen de manera administrativa, es decir, el administrador tendrá la oportunidad de crear estos equipos y, a la hora de realizar un enfrentamiento se hará un SELECT y se seleccionará aleatoriamente un número de todos los registros que haya y sacará la información de las ID's de los Pokémon que haya registrada para usar en combate. Tendrá estadísticas internas.

8. DESARROLLO DE PERFILES DE USUARIOS / ROLES.

En esta aplicación se usarán cinco niveles distintos de perfiles de usuarios y dos niveles distintos de administración.

- » Los perfiles de usuarios estarán divididos en cinco:
 - › Nivel gratuito, en base de datos no está representado ya que no será necesario tener una cuenta registrada y, por ende, no será necesario la utilización de la conexión a la base de datos. Tendrá acceso al apartado de acceso gratuito, a visualizar la página web y a enviar solicitudes de soporte en la aplicación.

Primera y segunda entrega.

- › Nivel básico, en base de datos estará representado como nivel “1”. En este nivel se tendrá acceso a lo anterior más dos ventanas de funcionalidades, como sería ver la pokedex dividida en regiones / juegos y acceso a los diferentes movimientos disponibles por generación.
- › Nivel Medio, en base de datos estará representado como nivel “2”. En este nivel se tendrá acceso a lo anterior más dos ventanas de funcionalidades, como sería ver los tipos que se han ido añadiendo por generación y las distintas Pokeballs que se han ido introduciendo en la aplicación. Además, se pueden crear hasta un máximo de cinco equipos Pokémon.
- › Nivel Alto, en base de datos estará representado como nivel “3”. En este nivel se tendrá acceso a lo anterior más tres ventanas de funcionalidades, como serían ver las bayas, los ítems de estado y los ítems de evolución. Además, se pueden tener enfrentamientos contra la IA.
- › Nivel Total, en base de datos estará representado como nivel “4”. En este nivel se tendrá acceso a lo anterior más tres ventanas de funcionalidades, como serían ver los ítems que se le pueden dar al Pokémon legendario “Arceus” para cambiar su tipo, los tipos de cartas que hay en el videojuego y las vitaminas que se pueden utilizar. Además, se podrá tener acceso a un ranking de los enfrentamientos que haya tenido.
- » Los perfiles de administradores estarán divididos en dos ya que no se contabiliza el que no sea administrador ya que su valor será “0”.
 - › El nivel básico, administrador. En base de datos estará representado como “1” a nombre de Administrador. Tendrá acceso a varias opciones en relación a los usuarios, podrá prohibir Pokémon para que no se incluyan a los equipos de Pokémon de los jugadores y podrán visualizar los tickets de soporte. Añadiendo una última cosa a lo anterior, las acciones que estos administradores hagan serán en todo momento registradas, para llevar un mejor control de sus acciones.
 - › El nivel más alto, es super Administrador. En base de datos estará representado como “2” a nombre de Super Administrador. Tendrá acceso a lo anterior más

Primera y segunda entrega.

el tener habilitado el control de niveles de usuarios y/o cambios de nombre que un usuario pueda pedir, además, si se llega a obtener este rango de super administrador, no se registrarán logs referentes a él.

9. DESARROLLO DE CONCURRENCIA EN LA APLICACIÓN.

El desarrollo de concurrencia que se llevará a cabo en el funcionamiento de la aplicación es, por el momento, de dos formas distintas:

- » Al iniciar la aplicación. Se prevé crear un hilo nada más ejecutar la aplicación para hacer las consultas a la API que más tiempo requieren para cargar, haciendo así que a la hora de usarlas sea más rápido y ágil el sacar los datos necesarios. Por ejemplo, para el apartado “Pokedex” que estará disponible para todos los usuarios, al elegir una opción u otra para mostrar se demora demasiado en cargar todos los datos disponibles ya que, tal vez, por apartado haya un mínimo de ciento cincuenta registros con más de diez mil líneas de *json* y, si se metiesen en una lista únicamente los datos que necesito, se cargarían más rápidos los datos, haciendo únicamente una llamada para los datos de un solo registro y sus evoluciones.
- » Al entrar al apartado de enfrentamientos. La idea es hacer un hilo principal que maneje el tempo del combate en reglas generales, es decir, un cronómetro que dicte cuando acaba el tiempo que se dispone y se queden empates en caso de que se acabe. Dentro de este hilo principal se buscaría ir manejando otro secundario, que maneje el tiempo del turno del jugador, es decir, si es el turno del jugador “1” y tiene un tiempo de treinta segundos para elegir que hacer y se le acaba, el turno se dará por finalizado y no podrá atacar al rival, pasando al ataque del rival y reiniciando ese hilo mientras que el hilo principal está en ejecución.

10. DESARROLLO EMPRESARIAL.

No me ha dado tiempo para esta entrega.