



# <POKÉDEXPLORER>

= "Exploración de la Pokédex" </POKÉDEXPLORER>

Tutor: Pablo Lledó Martínez

**Alba Marco Checa**

*Desarrollo de aplicaciones multiplataforma*

## 1. ÍNDICE.

|   |    |
|---|----|
| 1. ÍNDICE.....  | 1  |
| 2. ABSTRACT.....  | 4  |
| 2.1. ABSTRACT CASTELLANO.....   | 4  |
| 2.2. ABSTRACT INGLÉS.....   | 6  |
| 3. JUSTIFICACIÓN DEL PROYECTO.....                                      | 9  |
| 4. INTRODUCCIÓN.....  | 11 |
| 5. OBJETIVOS.....   | 14 |
| 6. DISEÑO.....  | 15 |
| 6.1. BASE DE DATOS.....   | 15 |
| 6.2. ELEMENTOS DE LA INTERFAZ GRÁFICA.....                              | 23 |
| 6.3. INTERFAZ GRÁFICA WPF.....  | 28 |
| 6.4. INTERFAZ GRÁFICA HTML Y CSS.....                                   | 32 |
| 6.5. DIAGRAMAS DE USO.....  | 38 |
| 7. DESARROLLO.....  | 41 |
| 7.1. CREACIÓN, CONFIGURACIÓN E IMPLEMENTACIÓN DE LA BASE DE DATOS.....  | 41 |
| 7.2. DESARROLLO DE LA INTERFAZ E IMPLEMENTACIÓN DE FUNCIONALIDADES..... | 43 |
| 7.3. PRUEBAS UNITARIAS Y AJUSTES Y MEJORAS.....                         | 44 |
| 7.4. DESARROLLO DE PERFILES DE USUARIO / ROLES.....                     | 44 |
| 7.5. DESARROLLO DE CONCURRENCIA EN LA APLICACIÓN.....                   | 46 |
| 7.6. DESARROLLO DE SEGURIDAD EN LA APLICACIÓN.....                      | 47 |
| 8. DESARROLLO EMPRESARIAL.....  | 48 |
| 8.1. PRESENTACIÓN DEL PROYECTO Y DEL EMPRENDEDOR.....                   | 48 |
| A. PRESENTACIÓN BREVE DEL PROYECTO.....                                 | 48 |
| B. INTERÉS Y CAPACIDAD EMPRENDEDORA.....                                | 48 |

|      |   |    |
|------|---|----|
| 8.2. | IDEA DE NEGOCIO.....  | 49 |
| A.   | REDACCIÓN DE LA IDEA Y PROPUESTA DE VALOR Y OBJETIVOS.....            | 49 |
| 8.3. | MERCADO Y CLIENTES. ....  | 49 |
| A.   | TIPO DE MERCADO.....  | 49 |
| B.   | SEGMENTO DE MERCADO. ....   | 49 |
| C.   | ESTUDIO DE MERCADO: LOS CLIENTES. ....                                | 50 |
| 8.4. | ENTORNO Y COMPETENCIA.....  | 50 |
| A.   | ENTORNO GENERAL Y ESPECÍFICO.....                                     | 50 |
| B.   | COMPETENCIA.....  | 50 |
| C.   | ANÁLISIS DAFO. ....   | 51 |
| D.   | LOCALIZACIÓN DE LA EMPRESA. ....                                      | 52 |
| E.   | MISIÓN, VISIÓN Y VALORES. ....  | 52 |
| F.   | RESPONSABILIDAD SOCIAL CORPORATIVA.....                               | 52 |
| 8.5. | ESTRATEGIAS DE MARKETING. ....  | 53 |
| A.   | ESTRATEGIA DE POSICIONAMIENTO, PRODUCTO, PRECIO, DISTRIBUCIÓN. ....   | 53 |
| B.   | ATENCIÓN AL CLIENTE. ....   | 54 |
| 8.6. | RECURSOS HUMANOS. ....  | 54 |
| A.   | ORGANIGRAMA DE LA EMPRESA. ....                                       | 54 |
| B.   | DISEÑO PUESTOS Y SELECCIÓN DE PERSONAL.....                           | 54 |
| C.   | MOTIVACIÓN LABORAL.....   | 55 |
| D.   | CONTRATACIÓN Y COSTE TRABAJADORES.....                                | 55 |
| E.   | ORGANIZACIÓN DE LA PREVENCIÓN.....                                    | 55 |
| 8.7. | FORMA JURÍDICA. ....  | 56 |
| A.   | FORMA JURÍDICA Y MOTIVOS. ....  | 56 |
| 8.8. | PLAN DE PRODUCCIÓN.....   | 56 |
| A.   | PLAN PRODUCCIÓN Y PROVEEDORES   GESTIÓN DE ALMACÉN E INVENTARIOS..... | 56 |

|       |   |    |
|-------|---|----|
| 8.9.  | ANÁLISIS ECONÓMICO - FINANCIERO. ....   | 56 |
| A.    | ANÁLISIS DE COSTES [FIJOS, VARIABLES, UMBRAL RENTABILIDAD Y PRECIO DE VENTA]..... | 56 |
| B.    | INVERSIONES Y GASTOS INICIALES.....   | 57 |
| C.    | PLAN FINANCIACIÓN. ....   | 57 |
| D.    | PREVISIÓN DE RESULTADO Y BALANCE DE SITUACIÓN.....                                | 57 |
| E.    | PLAN TESORERÍA. ....  | 57 |
| F.    | ESTIMACIÓN DE IMPUESTOS. ....   | 58 |
| 8.10. | PLAN DE PUESTA EN MARCHA.....   | 58 |
| A.    | TRÁMITES Y PLAZOS Y DOCUMENTACIÓN ADMINISTRATIVA - CONTABLE.....                  | 58 |
| 9.    | Conclusiones.....   | 59 |
| 10.   | Bibliografía. ....  | 60 |
| 11.   | Anexos.....   | 61 |

## 2. ABSTRACT.

### 2.1. ABSTRACT CASTELLANO.

El objetivo general de este proyecto es tener a mano una serie de información que sea más sencillo de acceder y esté en un lugar que no estar de página en página mirando lo que se busca sobre el tema elegido, en este caso, Pokémon. En conclusión, se busca implementar todo el contenido de información de, por ejemplo, *Pokémons* u objetos en un mismo lugar, navegando por pestañas, sin carga ninguna y en un mismo lugar.

Para poder llevar a cabo este proyecto se han utilizado varias herramientas distintas, desde en línea a escritorio.

Primeramente, se llevó a cabo una investigación de las diferentes páginas webs que llevan a cabo la recopilación de datos para estudiar su funcionamiento y su enlace a diferentes datos para ver como trabajaban.

Una vez terminada dicha investigación se miró las herramientas que se podrían utilizar en desarrollo y que ofrecían.

Para empezar, se ha utilizado el sistema que la herramienta GIT te ofrece, para poder tener todos los datos del proyecto siempre en sincronización, además de la implementación de la misma para los IDE's utilizados, haciendo más fácil la gestión de versiones entre todo el proyecto. Como principal IDE Microsoft Visual 2022 (IDE Microsoft Visual Studio, s.f.), se creó una solución con dos proyectos, uno utilizando WPF donde se llevó a cabo la aplicación y un proyecto de SandCastle para que la documentación sea más fácil de realizar y que sea un proceso más automatizado con los comentarios hechos en código.

Otra herramienta utilizada es el IDE Visual Studio Code 2022 (Visual Studio Code, s.f.) para la creación de la página web que se utiliza en complementación a la aplicación. En ella se ha llevado a cabo una estructura usando HTML, CSS y PHP para el acceso a base de datos. Para llevar a cabo su subida a un hosting de páginas web se ha utilizado una herramienta FTP, en este caso WinSCP (WinSCP, s.f.) por su uso previo. Además, antes de encontrar un hosting adecuado para el proyecto se llevó las pruebas en XAMPP para comprobar el uso de PHP y conexiones a base de datos. El hosting elegido ha sido "InfinityFree", que ofrecía un dominio y la utilización de PHP y FTP.

La base de datos del proyecto está hosteado en otro lugar totalmente gratuito llamado “db4free”. Para su gestión utiliza lo mismo que utiliza XAMPP, *phpmyadmin*, por lo que ha sido bastante familiar su uso y rápido adaptarse para la gestión. Además, utiliza un lenguaje común, que es SQL.

Continuando las herramientas, en el uso de la aplicación WPF se ha tenido que modificar imágenes para que encajase con lo que se quería conseguir, para ello se ha utilizado todo herramientas gratuitas y de libre uso en páginas webs tanto para quitar fondos, editar, voltear y demás funciones necesarias para lograr el objetivo. Una de la primaria para poder crear los títulos que se deseaban con el tipo de fuente es el siguiente: “<https://www.fontspace.com/pocket-monk-font-f23540>”. La siguiente que se utilizaba era un editor, parecido a la famosa herramienta Photoshop, pero totalmente en línea y gratuita llamada “<https://www.photopea.com/>”, para poder poner un contorno, con los colores de Pokémon o para quitar fondo a las distintas imágenes utilizadas en la aplicación. Para poder saber que colores utilizar se utilizó la herramienta “*Instant Eyedropper*” que te ayuda con la identificación de los colores según donde se encuentre el cursor, dándote un código RGB. Estos códigos RGB se utilizaban posteriormente en la página “<https://encycolorpedia.es/>”, que te ayuda para ver los diferentes todos que tiene ese color y ver variaciones. Para saber la paleta de colores que sigue la página de Pokémon, se ha utiliza la información otorgada por la siguiente página “[https://www.wikidex.net/wiki/Ayuda:Paleta\\_de\\_colores](https://www.wikidex.net/wiki/Ayuda:Paleta_de_colores)”.

Para el uso de la API y la lectura posterior de los JSON que se reciben, se han utilizado 3 páginas web. La primera es la misma API, utilizando la llamada “*PokéAPI*”. Para ver y utilizar correctamente su estructura de JSON se han utilizado dos herramientas, una siendo más visual viendo las relaciones desde el principio hasta el final: “*JSONCrack*”, y para ver la estructura que tiene el JSON de forma más simple y sólo estructurada hacia abajo (aunque se puede utilizar otros modos que se ofrecen en la página.) se ha utilizado “*Postman*”.

Se han llevado a cabo la implementación de técnicas como Modelo – Vista – Controlador, organizando de manera arbolada la aplicación para tener por un lado la parte del

código Modelo donde ocupa las imágenes en un archivo de recursos y el código de base a datos.

La parte de la vista, donde estarán las diferentes ventanas que se han utilizado, ordenadas correctamente según sea, por ejemplo, si son parte de la ventana de administración, se organizará en una carpeta dentro la ventana de administración.

La parte del controlador que hay detrás de cada ventana, para llevar a cabo la comunicación entre los botones y el código y poder actualizar así la ventana que el usuario esté viendo con la información que vaya añadiendo o actualizando al pulsar botones u otros elementos.

Otra técnica utilizada es la gestión de errores dentro de la aplicación, para evitar que la aplicación se pare o errores que afecten negativamente al usuario e interfiera con su experiencia. Para ello se ha tenido que ir haciendo pruebas poco a poco por trozos de aplicación, es decir, pruebas unitarias para ver donde ocurría y poner una solución.

Aparte de las mencionadas, se utiliza un desarrollo basado en componentes, es decir, unas ventanas llamadas "UserControl" que son portables y, una vez desarrolladas, se pueden llevar y utilizar en otras aplicaciones, consiguiendo que sea un código reutilizable.

Por último, se ha implementado conceptos importantes para el desarrollo de software como sería la concurrencia y la programación segura para conseguir una aplicación la cuál no se bloquee y siga trabajando en segundo plano para los procesos más complejos y lentos y para evitar "SQL Injection" entre otros. Mejorando así tanto la seguridad como el rendimiento de la aplicación y asegurar una integridad en los datos que se utilicen en la misma.

## **2.2. ABSTRACT INGLÉS.**

The overall objective of this project is to have at hand a series of information that is easier to access and is in one place than being from page to page looking for what you are looking for on the chosen topic, in this case, Pokémon. In conclusion, it seeks to implement all

the information content of, for example, Pokémons or objects in one place, browsing by tabs, without any load and in one place.

In order to carry out this project, several different tools have been used, from online to desktop.

Firstly, an investigation of the different web pages that carry out data collection was carried out to study how they work and how they link to different data to see how they work.

Once this research was completed, we looked at the tools that could be used in development and what they offered.

To begin with, we used the system that the GIT tool offers you, to be able to have all the project data always in synchronization, in addition to the implementation of the same for the IDE's used, making easier the version management between the whole project. As the main IDE Microsoft Visual 2022, a solution was created with two projects, one using WPF where the application was carried out and a SandCastle project to make the documentation easier and to make it a more automated process with the comments made in code.

Another tool used is the Microsoft Visual Studio 2022 IDE for the creation of the web page used to complement the application. A structure has been created using HTML, CSS and PHP for database access. To upload the files to a web hosting, a FTP tool has been used, in this case WinSCP because of its previous use. In addition, before finding a suitable hosting, tests were carried out in XAMPP to check the use of PHP and database connections. The hosting chosen was "InfinityFree", which offered a domain and the use of PHP and FTP.

For the database we used a hosting called "db4Free", which uses phpmyadmin as database management and uses SQL.

Following the tools used in the WPF application, it has been necessary to modify images to fit with what we wanted to achieve, for this we have used all free tools and of free use in web pages. One of the first to be able to create the titles that were desired with the type of source is the following: "<https://www.fontspace.com/pocket-monk-font-f23540>". The next one that was used was an editor, similar to the famous Photoshop tool, but totally online and free called "<https://www.photopea.com/>", to be able to put an outline, with the colors of Pokémon or to remove background to the different images used in the application. In order



to know which colors to use, the "[Instant Eyedropper](#)" tool was used, which helps you with the identification of the colors according to where the cursor is located, giving you an RGB code. These RGB codes were later used in the "<https://encycolorpedia.es/>" page, which helps you to see all the different colors of that color and to see variations. To know the color palette that follows the Pokémon page: "[https://www.wikidex.net/wiki/Ayuda:Paleta de colores](https://www.wikidex.net/wiki/Ayuda:Paleta_de_colores)".

For the use of the API and the subsequent reading of the JSON received, 3 web pages have been used. The first one is the API itself, using the one called "[PokéAPI](#)". To see and use correctly its JSON structure two tools have been used, one being more visual seeing the relations from the beginning to the end: "[JSONCrack](#)", and to see the structure that has the JSON in a simpler way and only structured downwards (although you can use other modes offered on the page.) "[Postman](#)" has been used.

We have carried out the implementation of techniques such as Model - View - Controller, organizing in a tree-like manner the application to have on one side the Model part of the code where it occupies the images in a resource file and the database code.

The part of the view, where will be the different windows that have been used, ordered correctly according to it, for example, if they are part of the administration window, it will be organized in a folder inside the administration window.

The part of the controller behind each window, to carry out the communication between the buttons and the code and to be able to update the window that the user is viewing with the information that is being added or updated by pressing buttons or other elements.

Another technique used is error management within the application, to avoid application crashes or errors that negatively affect the user and interfere with their experience. For this purpose, it has been necessary to test little by little by application pieces, that is. unit tests to see where it happened and provide a solution.

Apart from those mentioned above, a component-based development is used, that is windows called "UserControl" that are portable and, once developed, can be carried and used in other applications, making it a reusable code.

Finally, we have implemented important concepts for software development such as concurrency and secure programming to achieve an application which does not crash and continues working in the background for the most complex and slow processes and to avoid "SQL Injection" among others. Thus improving both the security and performance of the application and ensuring the integrity of the data used in it.

### **3. JUSTIFICACIÓN DEL PROYECTO.**

La justificación de que este proyecto haya surgido es el problema que tienen las páginas de información. Cuando alguien comenta cualquier tipo de información en su página web suele ser una página enfocada a foros y/o información de noticias nuevas que vayan saliendo del mundo. Además, las páginas de información más orientada al estilo “*wikipedia*” suelen pecar de no tener un índice, por lo que la búsqueda puede no ser demasiado clara y poco amigable con la experiencia de usuario, dificultando que su experiencia y siendo negativa. Si se investigan más páginas, la mayoría son páginas de interés general, como puede ser “*Vandal*”, una página que bien puede tener información sobre Pokémon o bien te puede comentar que juegos nuevos van a ser tendencia, es decir, coge varios temas distintos y no es especializada. Con esta aplicación se busca la solución de juntar esa misma información de diferentes espacios, páginas webs y demás foros en un mismo lugar.

¿Cómo se puede llegar a dar una solución de esa manera? Haciendo una aplicación de ámbito general sobre el mundo Pokémon, quitando la parte de noticias nuevas y facilitando al usuario información sobre los Pokémon, sobre sus habilidades, sobre sus tipos, sobre sus movimientos y demás datos que haya disponible para visualizar. De esta manera otorgará al usuario una vista más compacta sin tener que andar por demasiados sitios, facilitando esa búsqueda y ahorrando el tiempo que se podría perder si tiene que ir a distintas páginas para buscar lo que quiere.

Los objetivos, en primera instancia, será la búsqueda de un Pokémon por número en pokedex nacional o por su nombre. En segunda instancia, será la información de las distintas Pokedex, los movimientos por generación y sus estadísticas, los tipos por generación y sus debilidades / fortalezas, los tipos de Pokeballs que hay disponible y su origen, las bayas que hay con el efecto que hace, y ordenadas según categoría, objetos para cambiar un estado, ordenado también en cuatro tipos según lo que hace, los objetos necesarios para evolucionar un Pokémon, con a qué Pokémon afecta, las láminas disponibles que Arceus puede usar y con una descripción de lo que hacen, con los distintos tipos de carta y su costo y, por último, las distintas vitaminas que hay disponibles, con una descripción y coste.

Todo lo mencionado anteriormente tendrá una imagen para poder identificar su forma y saber que es dentro del juego y/o serie animada.

Además, se busca ampliar esta idea en futuros para poder añadir algo a favoritos y tener un fácil acceso, hacer equipos con distintos Pokémon y realizar combates con esos equipos.

Una vez completada la idea principal se buscará ampliar la información, ofreciendo más contenido ya sea gratuito o de pago, añadiendo opciones de resolución para el usuario, añadiendo mejoras tanto en la parte de carga de la información como en la parte de seguridad de la aplicación. Además, se irán añadiendo detalles pequeños para contentar al usuario y dar una mejor calidad e impresión en el uso.

Se considera que al lanzar la aplicación a mercado se tiene una oportunidad, aunque la visibilidad es baja debido a la función que hará, ahorrando el tiempo a los usuarios que juegan a este videojuego, ofreciendo que pierdan menos tiempo en búsqueda y más tiempo en disfrutar del videojuego. Aunque, viendo el punto malo de lanzar la aplicación es que sólo será de escritorio en primera instancia, haciendo que sea obligatorio el tener un sistema operativo que sea aceptado por la aplicación y, viendo las edades en las que se juega, puede ser que una gran mayoría de usuarios no disponga de un ordenador para poder hacer uso de la aplicación, por lo que se buscará aumentar la página web para que ofrezca la misma información que dentro de la aplicación de escritorio.

#### **4. INTRODUCCIÓN.**

Para comenzar con la introducción del proyecto, el problema que se va a abordar es la falta de accesibilidad de la información disponible de Pokémon en un mismo lugar, es decir, se quiere facilitar la forma de buscar la información y simplificar su manera de mostrarla hacia el usuario. Esto se llevará a cabo en una aplicación de escritorio la cuál será amigable para el usuario y fácil de utilizar y, posteriormente se desarrollará esta tecnología en plataformas móviles y página web.

Se está trabajando en el concepto de Pokémon debido a que es un videojuego que siempre ha estado en auge, gracias a su gestión y a sus comienzos. Nintendo ha sabido darle juego para que, a pesar de que la funcionalidad sea la misma, siga siendo uno de los juegos más vendidos. Al ser un concepto que está en ese momento, es importante realizar algo que ayude aún más a conseguir esa información necesaria para poder jugarlo y sea un poco más fácil al estar informado de cómo funcionan los diferentes objetos o para ver los Pokémon.

El problema, como se ha mencionado varias veces con anterioridad, es la resolución de la navegación entre distintas páginas web y el funcionamiento que estas tienen para conseguir la información que se quiere de una forma poco intuitiva para el usuario que está buscando dicha información. El objetivo principal es abordar esos temas de búsqueda de información, divididos en diferentes roles para los usuarios ofreciendo a cada rol un poco más que el anterior, siendo el último nivel lo más completo y que más información y/o funcionalidades ofrece.

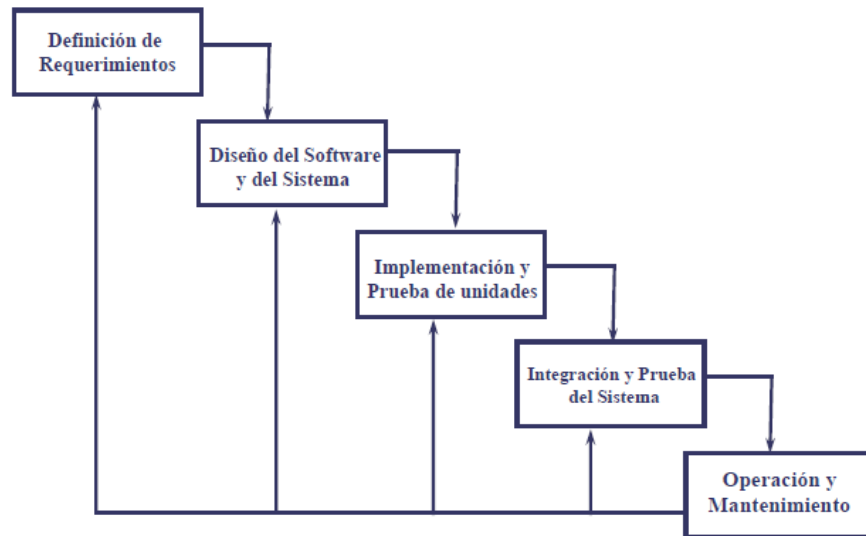
Por el momento la información que se está implementando será más básica, aumentando en el futuro más tipos diferentes de información. Entre la información más básica se encuentra la información del Pokémon (Tipos, movimientos que puede aprender, nombre, ID en pokedex Nacional, etc.) a excepción de las estadísticas básicas, ya que es un dato variable, todas las Pokedex que están actualmente disponibles, información de los tipos por generación, de los movimientos por generación, tipos de bayas, objetos de evolución, objetos de curación, los tipos de cartas existentes, las vitaminas y las láminas que se le pueden aplicar al Pokémon Arceus y sus efectos.

Para completar la información, se dispondrá de un apartado para el soporte técnico para los usuarios, para que así puedan informar de los errores que pueden llegar a tener o sugerencias.

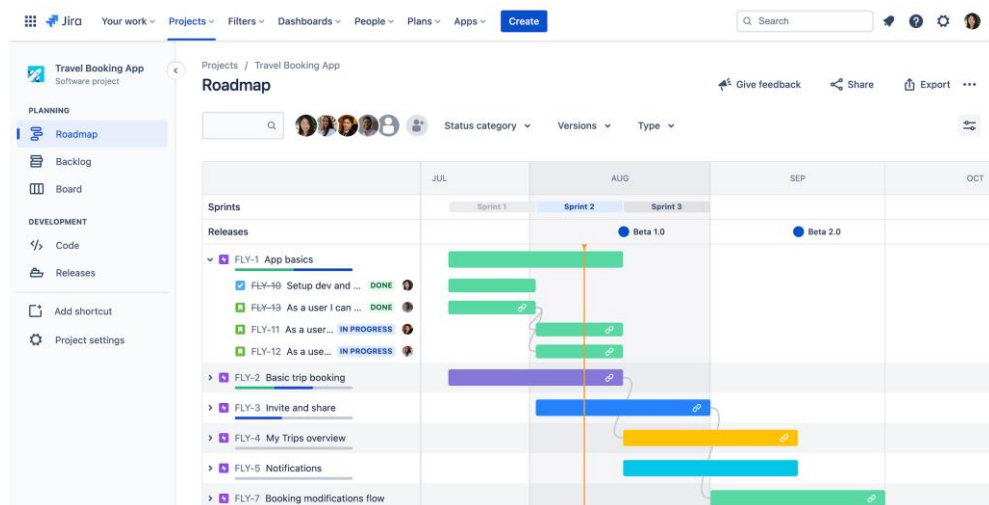
Además, se irá aumentando la aplicación para que no sea únicamente de investigación y recopilación de información, sino que sea de un ámbito mayor a todo lo que Pokémon ofrece como creación de equipos y enfrentamientos con esos equipos.

La metodología utilizada es el **modelo Cascada** (Metodología Cascada, s.f.), permitiendo así tener una valoración general de los conceptos que se quieren llevar a cabo durante este tiempo tanto en escritorio como en la página web. Permite tener el trabajo organizado y que esté detallado ya que está dividido en varias partes:

- Los requisitos. Para ello se tiene que valorar lo que se quiere llegar a hacer, en este caso la información.
- El análisis. Se ha de investigar a los rivales para ver cómo trabajan y, gracias a esa información, realizar un análisis de los resultados para saber que herramientas utilizar y como adecuarlas al trabajo de forma eficiente.
- Diseño. Se ha de definir que diseño se quiere llevar a cabo. En este caso se ha llevado un diseño parecido al que Pokémon ofrece, con su temática en toda la aplicación, tipos de fuentes e imágenes para poder ambientar mejor la aplicación y la página web a desarrollar.
- Programación. Se ha llevado a cabo una investigación de las posibilidades que ofrece C# para ver sus componentes en WPF y así saber cómo funciona correctamente y poder implementarlos en código. También se ha llevado una investigación del funcionamiento del PHP y su comunicación con C# para poder llevar la encriptación de datos correctamente desde la página web hacia la aplicación de escritorio.
- Pruebas. Realizar pruebas por bloques de código y corregir los errores, ya sea modificando su código o controlando la excepción, en este caso se controlan excepción como regla general.



Según el equipo de desarrollo de *PokeDEXplorer* aumente, se implementará una **metodología SCRUM** (Metodología Scrum, s.f.), usando *Atlassian* como principal plataforma para esta metodología debido a su funcionamiento en “sprints” de un periodo corto, perfecto para ir haciendo tareas y documentar su proceso de creación.



Hay que tener en cuenta en doble proyecto de back-end y front-end tanto en desarrollo web como en desarrollo escritorio, por ende, cuando se implemente la nueva metodología junto a un equipo, se deberá de llevar dos tableros diferentes, uno para cada tipo de programación para no llevar a malentendidos.

En la realización de este proyecto se busca lograr eficazmente los siguientes conceptos para los usuarios que utilicen la aplicación de escritorio.

- Mejorar la experiencia de usuario: Se busca mejorar la satisfacción que tengan los usuarios que han experimentado el buscar información en foros y/o páginas webs. Mejorando así su experiencia y haciendo que sea más fácil su uso y familiar de usar.
- Automatización de procesos: Internamente se busca que en la aplicación se pueda automatizar procesos tediosos o repetitivos, liberando tiempo para tareas más importantes y gestionar la memoria que usa, para que se use en otros apartados de la aplicación, mejorando así su rendimiento y la experiencia de usuario.
- Gestión eficiente de información: Se busca una planificación de los datos bien diseñada que ayude a ver en cada momento cada apartado para poder gestionar de una manera eficaz la cantidad de información que algún apartado tiene, mejorando así presión y la rapidez en generar un tipo de información u otro.
- Mayor escalabilidad: Se busca que la aplicación esté con un diseño principal que pueda ser adaptable tanto a futuros cambios como actuales, sin necesidad de tocar su estructura de manera abrupta para no alterar la experiencia de usuario. Además, se busca que su tamaño se pueda adaptar a todo tipo de pantallas y que sus componentes se vean adecuadamente.

## **5. OBJETIVOS.**

Se busca hacer una aplicación que sea útil a la hora de la búsqueda de información referente al mundo Pokémon, ya sea su videojuego o su serie animada. Además, se busca que su interfaz sea intuitiva y familiar para su uso y evitar que se usen páginas similares a lo que **PokeDEXplorer** quiere ofrecer y juntar todo en esta aplicación.

- Objetivo general.
  - › Desarrollar una aplicación de información de distintos tipos sobre el mundo Pokémon que permita añadir a favoritos y con un soporte técnico dentro de la aplicación, además de un sistema de administración funcional.
- Objetivos específicos.
  - › Definir los requerimientos funcionales que ha de tener de la aplicación, a partir de una revisión detallada de las páginas que ofrecen algo similar al objetivo.

- › Desarrollar la interfaz gráfica de usuario y de administración utilizando el framework WPF y el lenguaje de programación C#.
- › Implementar una base de datos para almacenar la información de los usuarios, utilizando el sistema de gestión de bases de datos phpMyAdmin del host.
- › Desarrollar la funcionalidad de registro en la página web para un posterior inicio de sesión en la aplicación de escritorio, utilizando un sistema de cifrado, evitar SQL Injection y aplicando más medidas de seguridad para poder asegurar la integridad de los datos.
- › Desarrollar una página web con temática de Pokémon para descargar la aplicación e información que pueda ser útil.
- › Implementar un sistema de soporte técnico o atención al cliente.
- › Implementar la adicción a favoritos de los distintos objetos que el usuario seleccione.
- › Implementación de Cinco niveles distintos de usuarios y Dos niveles distintos para administración.
- › Aplicar métodos de concurrencia en C# para la carga de API's pesadas.
- › Desarrollar los distintos apartados de información, haciendo así una experiencia más agradable para el usuario.
- › Realizar ejecutables para la instalación, descargables desde la página web.
- › Realizar pruebas exhaustivas de la aplicación para asegurar su calidad y corrección en el funcionamiento.
- › Realizar pruebas para evitar la ofuscación de código.
- › Ofrecer una documentación del código utilizado.

## **6. DISEÑO.**

### **6.1. BASE DE DATOS.**

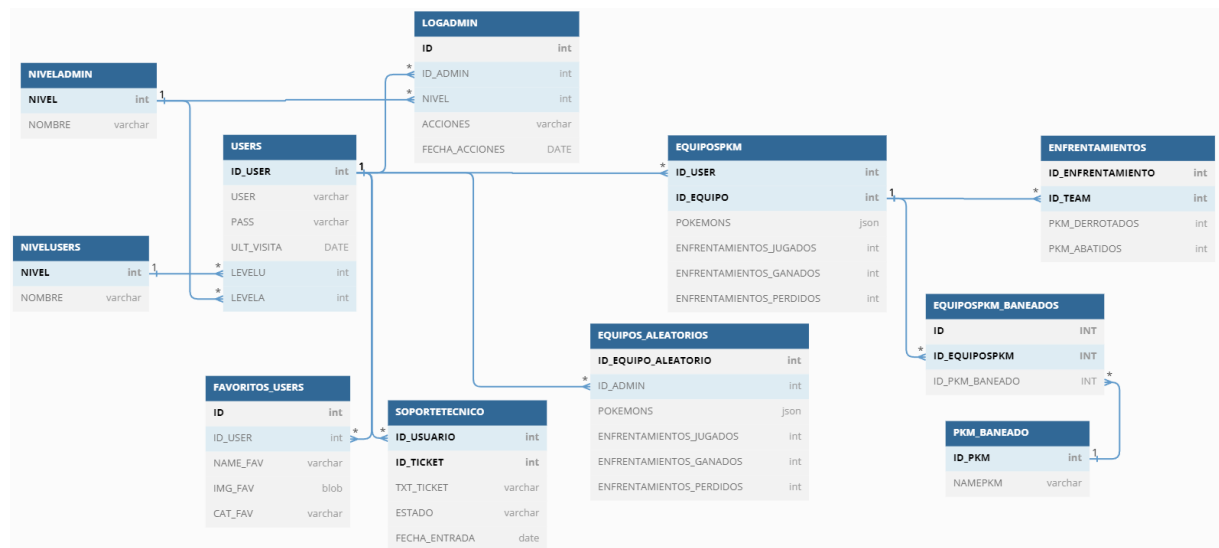
Para comenzar con el proyecto se ha de tener una base de datos, ya sea NO-SQL, Relacional, etc.



En este caso se ha aplicado el formato Entidad – Relación para toda la información que gira en torno a los usuarios y tener todo comunicado de la mejor manera.

Además de las tablas creadas para gestionar la información de los usuarios, están las tablas auxiliares siguiendo el formato Entidad – Relación, que son tablas necesarias para poder almacenar información sobre los tickets de soportes, saber que Pokémon se han baneado para no permitir su integración en los equipos y que quede registro de que intentó añadir un Pokémon baneado o para la creación de equipos para formar equipos para que los usuarios se puedan enfrentar a algún equipo pre hecho, registrando que administrador creó dicho campo.

*Diagrama con relaciones y campos de la base de datos.*



*Diagrama con relaciones y campos de la base de datos desde phpMyAdmin.*

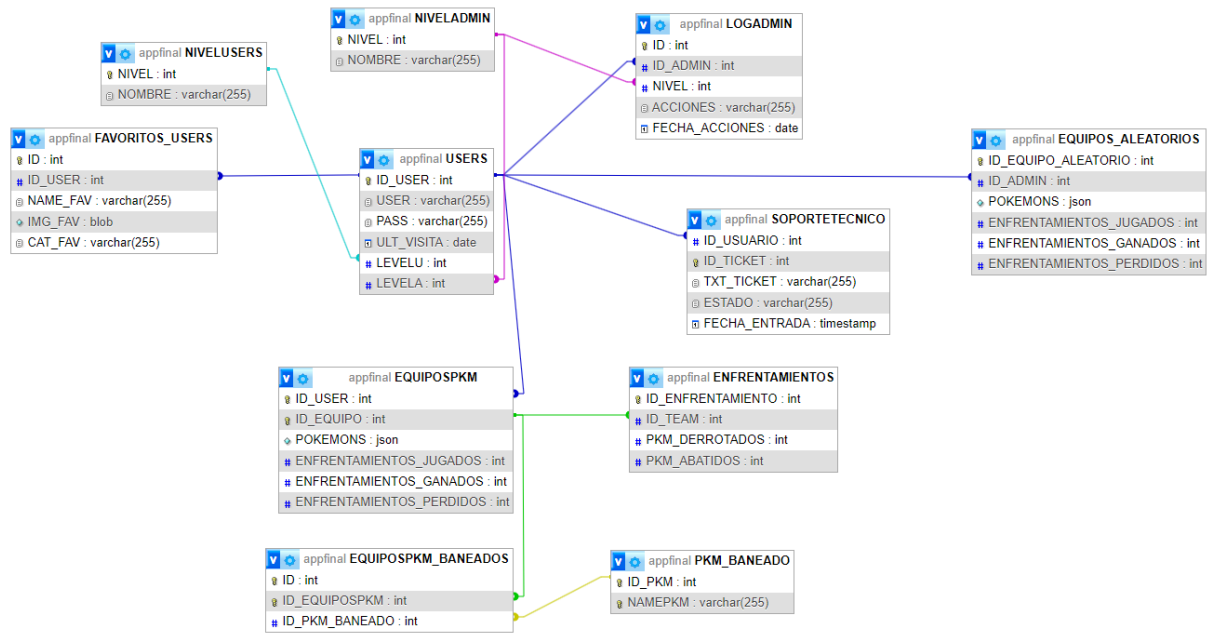
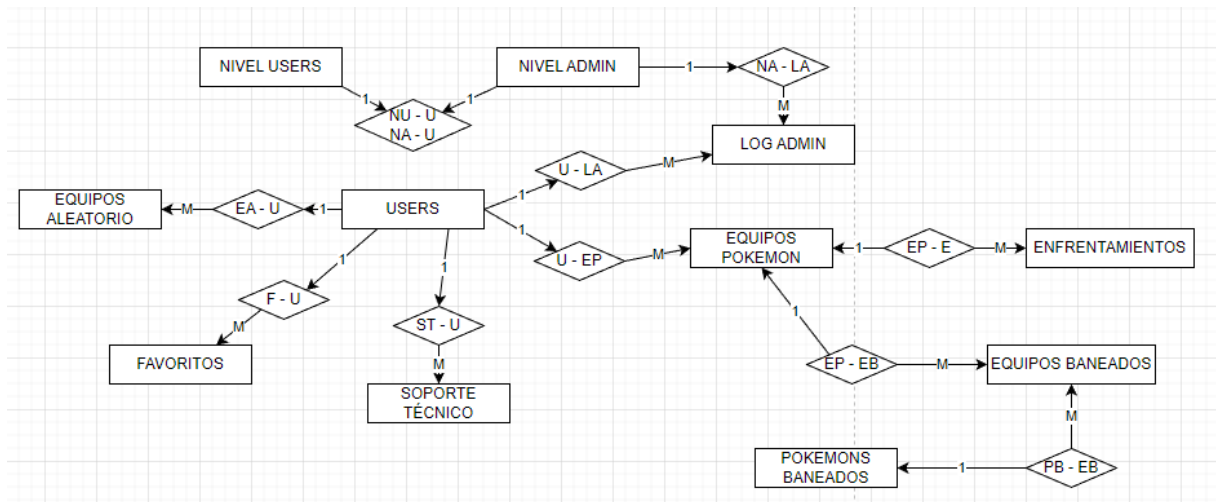


Diagrama conceptual de la base de datos.



En total, por el momento, se tiene 11 tablas, cada una para un propósito diferente.

(Documentación MySql y .NET, s.f.)

En primera instancia se explicará el funcionamiento de la tabla denominada “USERS”, ya que es el eje central del funcionamiento de la aplicación:

- » **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.

Utiliza un campo ID con autoincremento para ir registrando los usuarios y darle un identificador único. Los niveles son de tipo **int**, la última visita se almacena en formato americano, es decir, mm/dd/yyyy. Por último, los campos de nombre y contraseña son no nulos, es decir, se tienen que registrar si o si cuando se registre un 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* usando el sistema que PHP te proporciona de encriptación (Hash PHP, s.f.), el cual se puede usar a su vez en C# (Hash C#, s.f.).

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.

Además, dispone de dos tablas de donde sacará los datos del nombre del nivel de usuario y del nivel de administración.

Los datos de **nivel de usuario** serían los siguientes: {1, Básico; 2, Medio; 3, Alto; 4, Total;}

Los datos de **nivel de administración** serían los siguientes: {0, Nulo; 1, Admin; 2, Superadmin;}

De estas tablas se sacará el nivel, que está relacionado de uno a muchos hacia la tabla Users para obtener su nivel en todo momento. Si el nivel no existe a la hora de su creación, dará un error.

Como se ve tiene varias relaciones con otras tablas "LOGADMIN", "FAVORITOS\_USERS" y "EQUIPOSKMN".

| #                          | Nombre  | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|----------------------------|---|--------------|--------------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> 1 | ID_USER  | int          |                    |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | USER  | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 3 | PASS  | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 4 | ULT_VISITA  | date         |                    |           | Sí   | NULL           |             |                |
| <input type="checkbox"/> 5 | LEVELU   | int          |                    |           | Sí   | NULL           |             |                |
| <input type="checkbox"/> 6 | LEVELA   | int          |                    |           | Sí   | NULL           |             |                |

- **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.

Tiene un identificador único por cada registro nuevo que hay y se le asigna a un usuario ya existente con una relación de muchos a uno con la tabla USERS. El campo FECHA\_ACCIONES es de tipo DATE, con formato americano (mm/dd/yyyy) donde se almacenará la fecha con el tipo de dato con una fórmula de C# que saca mes, día y año, al igual que las demás exceptuando soporte técnico.

| #                          | Nombre         | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|----------------------------|----------------|--------------|--------------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> 1 | ID 🔑           | int          |                    |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | ID_ADMIN 🔑     | int          |                    |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 3 | NIVEL 🔑        | int          |                    |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 4 | ACCIONES       | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 5 | FECHA_ACCIONES | date         |                    |           | No   | Ninguna        |             |                |

- **FAVORITOS\_USERS** → En esta tabla se registrará cuando un usuario les dé 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 por su nombre y categoría, además, si tiene una imagen, se almacenará conjuntamente con un dato tipo blob para después leerlo y sacarlo en una ventana de favoritos.

Dependiendo de la categoría de lo que se guarde en la base de datos se mostrarán en un lugar u otro lugar de la pantalla de favoritos.

Al igual que las otras tablas tiene un campo de auto incremento que es el identificador interno del registro. Tiene una relación de muchos a uno con la tabla USERS.

| #                          | Nombre    | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|----------------------------|-----------|--------------|--------------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> 1 | ID 🔑      | int          |                    |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | ID_USER 🔑 | int          |                    |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 3 | NAME_FAV  | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 4 | IMG_FAV   | blob         |                    |           | Sí   | NULL           |             |                |
| <input type="checkbox"/> 5 | CAT_FAV   | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna        |             |                |

- **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.

Los Pokémon correspondientes se guardarán como tipo JSON que se crea en código según van eligiendo los diferentes Pokémon dentro de la aplicación, además, por cada vez que seleccionan un Pokémon se comprueba la tabla “PKM\_BANEADO”, para saber si está baneado o no lo está.

Tiene una doble relación, una de mucho a uno con la tabla “USERS” para saber a qué usuario corresponde dicho equipo y otra con la tabla “Enfrentamientos” de uno a muchos, porque un equipo creado puede tener cero, uno o muchos enfrentamientos, donde se irá actualizando según los Pokémon que vaya derrotando y le derroten.

En un futuro, desde esta tabla, se hará un ranking para saber qué equipo y de que usuario tiene un mejor porcentaje de jugados/ganados ordenando por el campo de enfrentamientos jugados y enfrentamientos ganados, junto a un sumatorio de la ID del usuario e ID del equipo.

| #                          | Nombre                   | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|----------------------------|--------------------------|------|--------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> 1 | ID_USER 🔑🔑               | int  |              |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 2 | ID_EQUIPO 🔑              | int  |              |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> 3 | POKEMONS                 | json |              |           | Si   | NULL           |             |                |
| <input type="checkbox"/> 4 | ENFRENTAMIENTOS_JUGADOS  | int  |              |           | Si   | NULL           |             |                |
| <input type="checkbox"/> 5 | ENFRENTAMIENTOS_GANADOS  | int  |              |           | Si   | NULL           |             |                |
| <input type="checkbox"/> 6 | ENFRENTAMIENTOS_PERDIDOS | int  |              |           | Si   | NULL           |             |                |

- » **ENFRENTAMIENTOS** → Se usará para saber con qué equipo se ha peleado, a los Pokémon que ha derrotado en general y a los Pokémon que le han matado en general, simulando a un contador. Además, se tendrá en cuenta la relación con la ID del equipo que ha llevado a cabo el enfrentamiento.

Si se repite dicho campo del ID del equipo, en vez de añadir registros nuevos, se hará un *update* con los resultados que se tengan en el nuevo enfrentamiento.

Tiene un campo de auto incremento para saber que enfrentamiento ha sido.

| #                        | Nombre              | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|--------------------------|---------------------|------|--------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> | 1 ID_ENFRENTAMIENTO | int  |              |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 ID_TEAM           | int  |              |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> | 3 PKM_DERROTADOS    | int  |              |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> | 4 PKM_ABATIDOS      | int  |              |           | No   | Ninguna        |             |                |

- SOPORTE\_TECNICO** → Esta tabla tiene relación a un usuario registrado, por lo tanto, el acceso al soporte técnico sólo podrá usuarios registrados por el momento.

Se ha de saber que usuario ha mando el reporte de los bugs y/o sugerencias para en las notas de la actualización nombrarle y agradecerle. Esta tabla únicamente porta información que se necesite en el campo TXT\_TICKET de tipo **varchar**, que en este caso sería el bug/sugerencia y que verá el administrador para dar solución. Los errores que se solucionen irán a la página web, en el apartado de Descargas conjuntamente a una nueva actualización.

Tiene un campo de autoincremento para ir insertados registros con un identificador único y tenerlos ordenados de más antiguo a más nuevo, para ir solucionándolos cuanto antes los más antiguo y así poder marcarlos cuanto antes de solucionado.

| #                        | Nombre          | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado    | Comentarios | Extra             |
|--------------------------|-----------------|--------------|--------------------|-----------|------|-------------------|-------------|-------------------|
| <input type="checkbox"/> | 1 ID_USUARIO    | int          |                    |           | No   | Ninguna           |             |                   |
| <input type="checkbox"/> | 2 ID_TICKET     | int          |                    |           | No   | Ninguna           |             | AUTO_INCREMENT    |
| <input type="checkbox"/> | 3 TXT_TICKET    | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna           |             |                   |
| <input type="checkbox"/> | 4 ESTADO        | varchar(255) | utf8mb4_0900_ai_ci |           | No   | Ninguna           |             |                   |
| <input type="checkbox"/> | 5 FECHA_ENTRADA | timestamp    |                    |           | No   | CURRENT_TIMESTAMP |             | DEFAULT_GENERATED |

- PKM\_BANEADO** → Es una tabla que únicamente tendrá acceso el grupo de administradores, independientemente de si es administrador o super administrador, 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 o el nombre de ese Pokémon seleccionado está en la tabla no permitirá su selección.

En caso de que se seleccione un Pokémon baneado se hará un registro nuevo en la tabla denominada "EQUIPOSPKM\_BANEADOS", seleccionando el ID\_PKM de esta tabla y seleccionando el ID\_EQUIPO de la tabla "EQUIPOSPKM".

| #                          | Nombre  | Tipo         | Cotejamiento       | Atributos | Nulo | Predeterminado | Comentarios | Extra |
|----------------------------|---------|--------------|--------------------|-----------|------|----------------|-------------|-------|
| <input type="checkbox"/> 1 | ID_PKM  | int          |                    |           | No   | Ninguna        |             |       |
| <input type="checkbox"/> 2 | NAMEPKM | varchar(255) | utf8mb4_0900_ai_ci |           | Si   | NULL           |             |       |

- EQUIPOSPKM\_BANEADOS** → Es una tabla con una relación en ambos campos, una para saber qué equipo seleccionó el equipo baneado y otro campo para saber el ID del Pokémon que se seleccionó. Su clave primaria es un ID con autoincremento en caso de que se intenten el mismo Pokémon para el mismo equipo.

| #                          | Nombre         | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|----------------------------|----------------|------|--------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> 1 | ID             | int  |              |           | No   | Ninguna        |             | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | ID_EQUIPOSPKM  | int  |              |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> 3 | ID_PKM_BANEADO | int  |              |           | No   | Ninguna        |             |                |

- 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 gracias al campo de autoincremento y sacará la información de las ID's de los Pokémon que haya registrada para usar en combate. Tendrá estadísticas internas, para a posterior sacar un ranking. Al igual que el apartado de "EQUIPOSPKM", la información de los Pokémon se almacenará en un **JSON**.

Sus campos funcionaran de la misma forma que se utilizan en la tabla de "EQUIPOSPKM", e internamente funcionarán de la misma manera, es decir, si hay un Pokémon baneado no lo dejará seleccionar y demás.

Tiene relación con el administrador que ha creado dicho equipo.

| #                          | Nombre  | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra |
|----------------------------|---|------|--------------|-----------|------|----------------|-------------|-------|
| <input type="checkbox"/> 1 | ID_EQUIPO_ALEATORIO  | int  |              |           | No   | Ninguna        |             |       |
| <input type="checkbox"/> 2 | ID_ADMIN             | int  |              |           | No   | Ninguna        |             |       |
| <input type="checkbox"/> 3 | POKEMONS  | json |              |           | No   | Ninguna        |             |       |
| <input type="checkbox"/> 4 | ENFRENTAMIENTOS_JUGADOS   | int  |              |           | Sí   | NULL           |             |       |
| <input type="checkbox"/> 5 | ENFRENTAMIENTOS_GANADOS   | int  |              |           | Sí   | NULL           |             |       |
| <input type="checkbox"/> 6 | ENFRENTAMIENTOS_PERDIDOS  | int  |              |           | Sí   | NULL           |             |       |

## 6.2. ELEMENTOS DE LA INTERFAZ GRÁFICA.

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”. Además, se ha utilizado en algunos casos una página que ayudaba con la saturación de un color o a decir colores complementarios y/o parecidos: [colores](#). Gracias a esta página se utilizan colores más detallados y que concuerdan de mejor manera con la temática que se está buscando en cuanto a colores.

Para los colores de letras se ha utilizado el color en HTML “Gold” y “Black” principalmente, utilizando algún otro color para alguna cosa puntual.

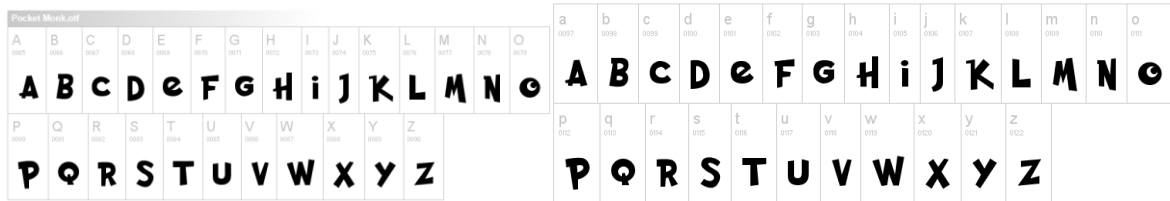
Para los estilos de fuentes se han implementado cuatro tipos distintos de letras estilo Pokémon, las cuales son:

*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.



- » **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.



(Descarga de fuentes, s.f.)

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]



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

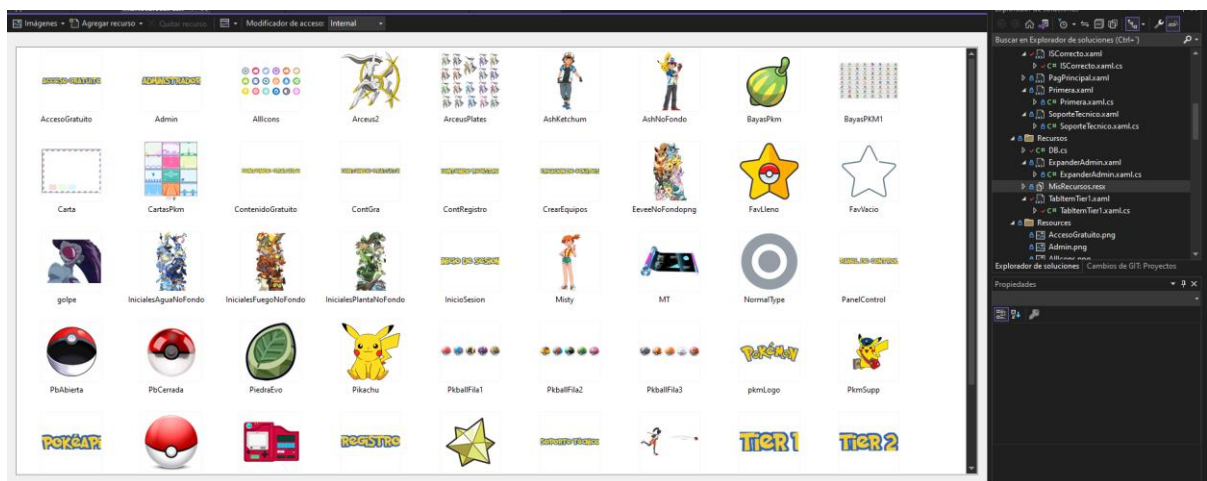
- » La última página que se utilizará es un **convertidor**. Este convertidor nos permitirá convertir la imagen hecha previamente con formato “psd” a formato “png” (hay disponibles más formatos, pero el utilizado en la aplicación es png para todas las imágenes). Es tan fácil como darle a “Convertir” 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.



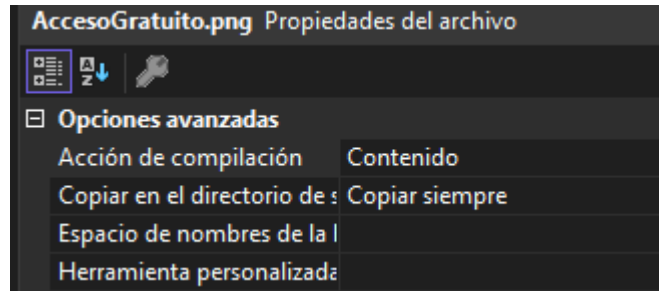


Para el almacenamiento de las imágenes se ha utilizado un recurso que .NET ofrece para el almacenaje de cadenas, datos binarios (como imágenes, iconos y clips de sonido) y objetos de programación para que pertenezcan con la aplicación a lo largo de su vida útil llamado **“Archivo de recursos”**. Hay que tener cuidado con estos archivos porque utilizan un formateador de serialización binaria para generar y analizar los datos binarios de un tipo especificado. Como resultado, un archivo **.resx** puede convertirse en un archivo no válido si el formato de serialización binaria de un objeto cambia de manera que sea incompatible.

Se puede crear tanto en código como visualmente como un componente. En mi caso ha sido creado de manera visual, en la carpeta de mi directorio **“Recursos”**.



A cada imagen que se haya ido guardando, para que pueda ser mostrada en el programa de forma correcta es necesario aplicarle la propiedad de **“Contenido”**, tal y como se muestra en la siguiente imagen.



De esta forma se mostrará en el programa al usar un objeto “Image” y, cuando se compile, se irán junto a los archivos al seleccionar “Copiar Siempre” en la opción de copiar en el directorio.

### 6.3. INTERFAZ GRÁFICA WPF.

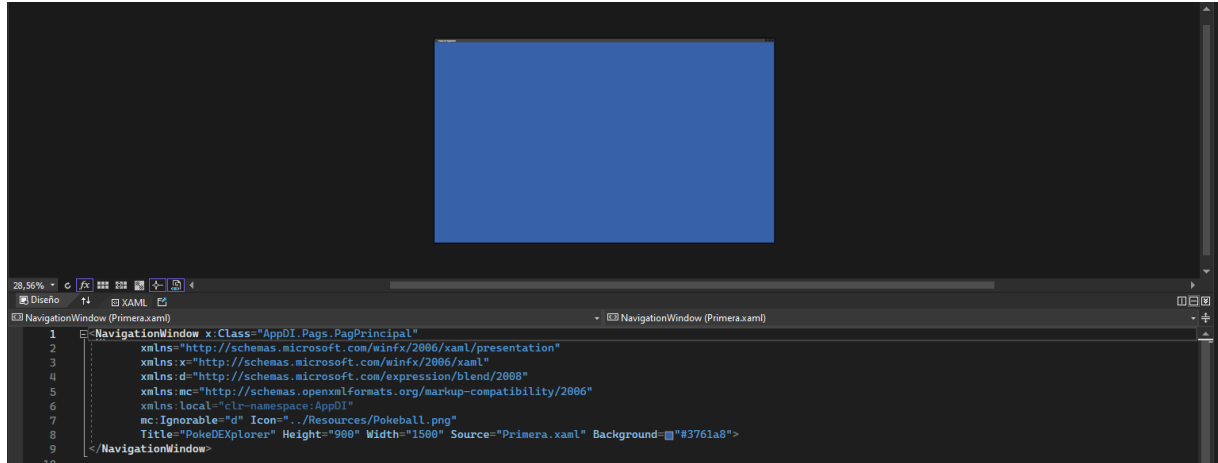
El diseño de base de datos que se ha utilizado está explicado en detalle en el punto anterior de Base de Datos, pero, en conclusión, la tabla maestra del programa es la de Users, la cual gira en torno al usuario.

El estilo de ventanas de la aplicación se hará de tipo Navegables (Documentación NavigationWindow, s.f.), 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 dónde 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.

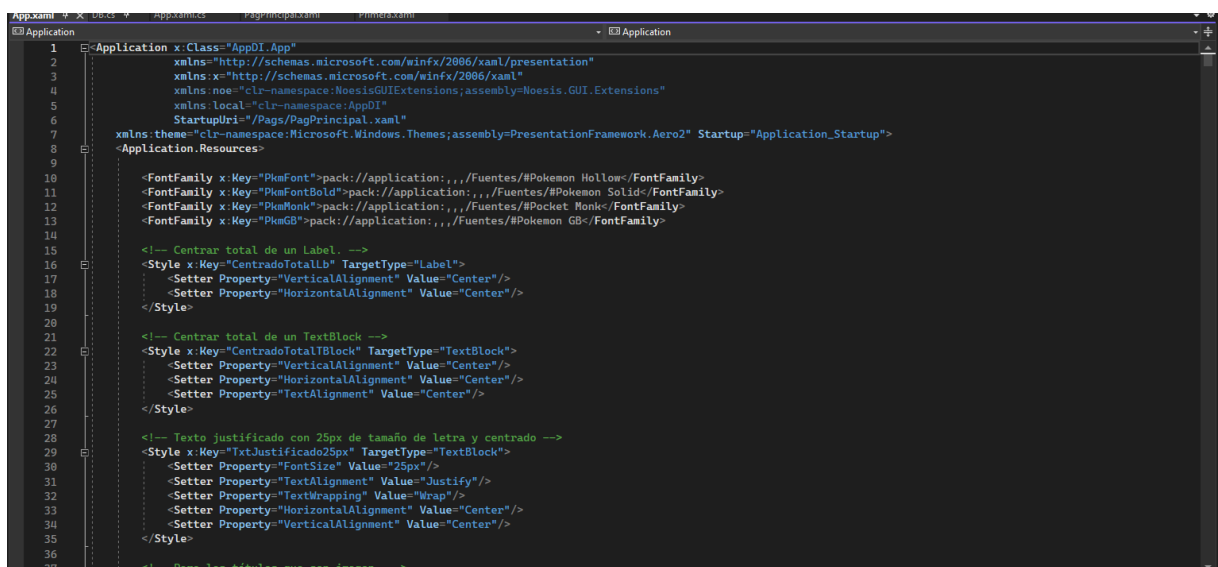
El tamaño por defecto del proyecto que se está llevando a cabo será de ancho 1500px y de alto 900px, haciendo así que sea más fácil de ver la letra al ser una ventana más grande. Por el momento no tiene resoluciones distintas a las mencionadas, pero es uno de los objetivos, hacer que la aplicación se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

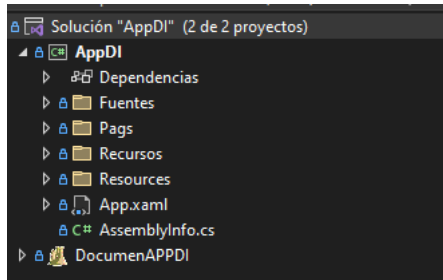
Para hacer funcionar el estilo de ventanas navegables se utiliza una ventana que es utilizada como contenedor principal de las demás ventanas, es decir, es como un padre donde sus hijos van yendo según lo asignado.



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 use 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.

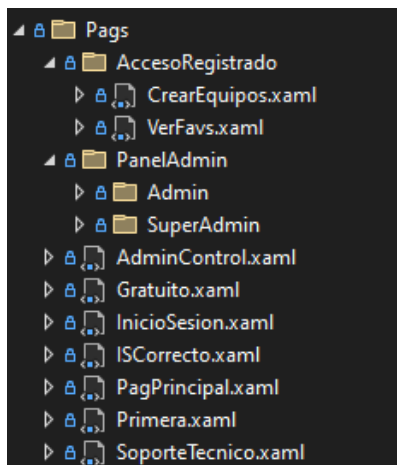
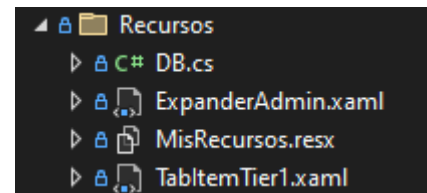




En la imagen presente en la parte izquierda del párrafo se ve como está organizado el proyecto.

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

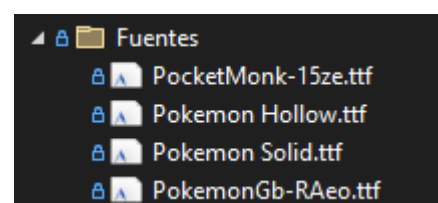
interacción entre la aplicación y lo que el usuario dice, con la parte servidor que utiliza para conectarse a la base de datos hosteada. Es la llamada parte “Modelo” ya que está la lógica y hacer que todo sea como un reloj a la hora de su funcionamiento.



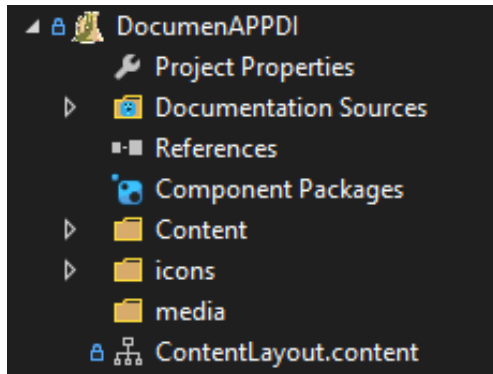
Dentro de la carpeta “Pags” está el apartado general en el cuál se tienen las ventanas de navegación y, dentro de las ventanas, un apartado referente al panel de administrador donde se almacena todas las funciones que estos tienen dependiendo si son administrador o si son super administradores y otro referente al contenido que se tiene si se usa una cuenta registrada.

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.



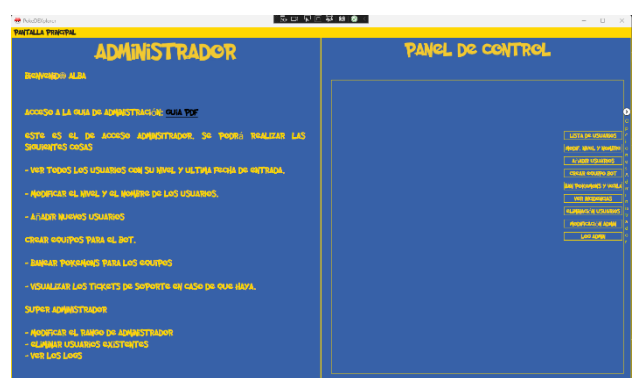
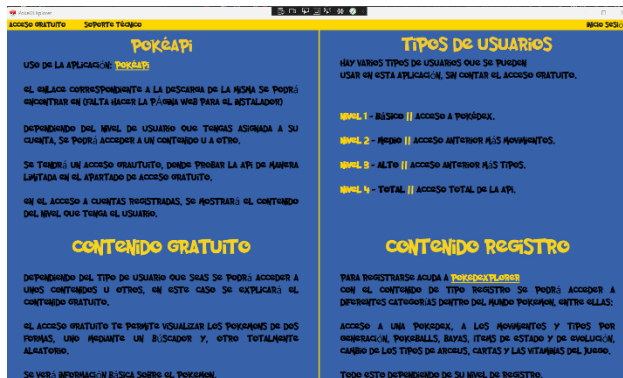
Por último, la carpeta de recursos que se utilizará para almacenar los recursos insertado en el archivo de “Recursos” → “MisRecursos.resx”. Es una carpeta que no se puede modificar ya que puede llevar a errores. Está explicado con más detalle [aquí](#).



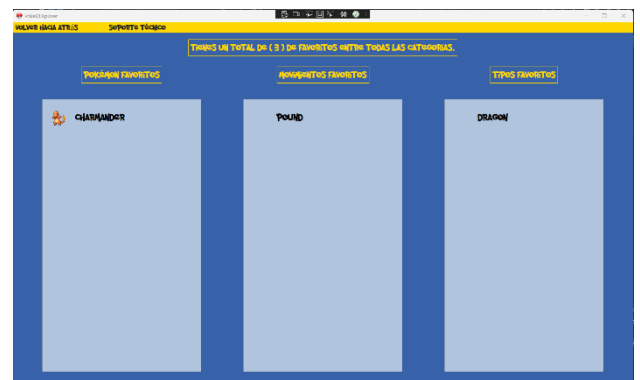
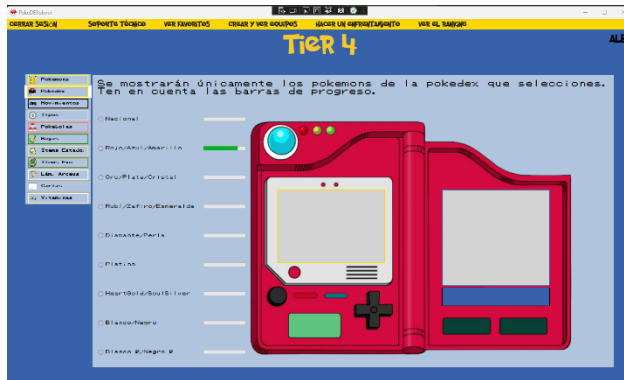
Otra parte de lo que se está realizando es la parte de documentación. En ella vendrá explicado los métodos realizados y demás información útil dentro del código para que se entienda mejor y sea más fácil saber su funcionamiento y su idea del porqué. Se utiliza una extensión llamada “Sandcastle” (Documentación SandCastle, s.f.), la cual te proporciona generar la

documentación a partir del código comentado.

Páginas de demostración de la propia aplicación:



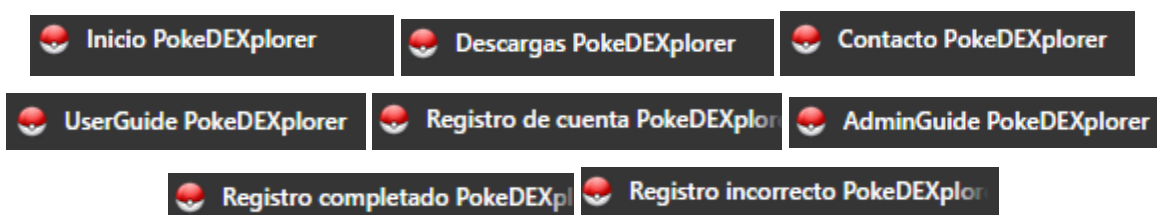




#### 6.4. INTERFAZ GRÁFICA HTML Y CSS.

Acceso a la página web: <http://pokeapialbamarco.42web.io/>.

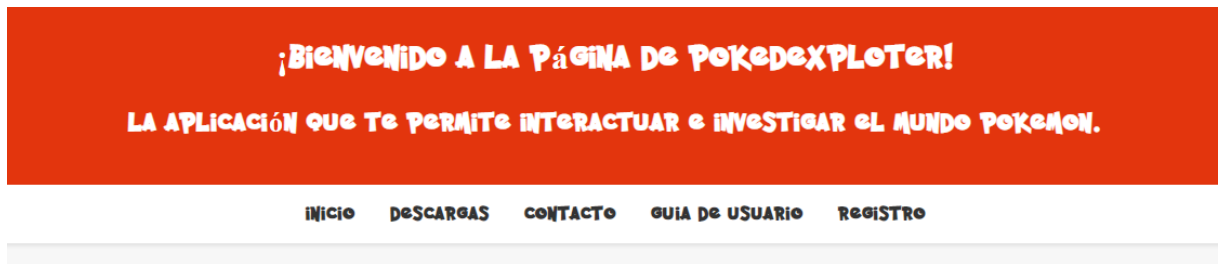
En la página web de **PokeDEXplorer** se ha llevado a cabo una interpretación sobre el estilo de Pokémon, siendo a su vez minimalista. Se ha buscado que sea una página accesible por todos los usuarios, además de que su uso sea intuitivo con una barra de navegaciones y la disposición de las páginas, es decir, dependiendo de en qué página se esté, el título de la pestaña irá variando y contendrá el icono de la Pokeball, además de la paleta de colores utilizados, estilo general de elementos y las imágenes.



Añadir a lo comentado con anterioridad es que se ha buscado la optimización de las páginas para que su velocidad aumente, además de implementar código seguro a la hora de registrarse y que no permita hacer SQL Injection (Param PHP, s.f.) y la adaptación de pantallas para diferentes dispositivos y tamaños de pantallas distintos, usando la medida que permite el diseño responsivo (Diseño Responsivo, s.f.).

Para que sea claro de leer se ha buscado que las letras color oro tengan un contorno azulado, al igual que el título original de "Pokémon" para que sea más familiar a los usuarios

y más legible sobre un fondo blanco. Además, los colores principales son blanco y negro, dependiendo del fondo en el que se encuentre para hacerlo parecer a una Pokeball.

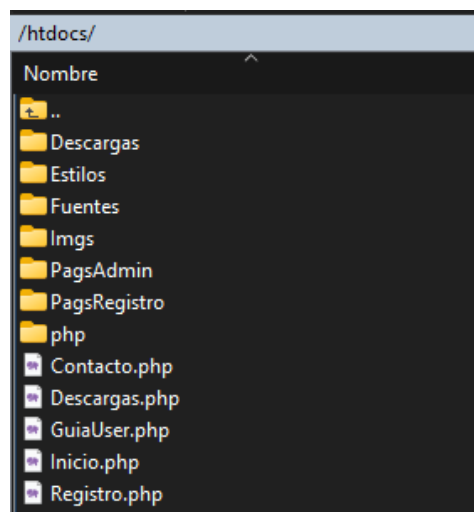


## BIENVENIDO AL MUNDO DE LOS POKÉMON

A lo largo de la página web, se irán mostrando alguna imagen / gif para rellenar los huecos vacíos y darle algo más de vida a la página, haciéndola más visual, algunos ejemplos son:



Todas las páginas están creadas con PHP, para poder tener su implementación en código ya que la parte de la fecha se hace con PHP para el año actual. En todas las ventanas se dispondrá de una barra de navegaciones. Se ha buscado realizar un directorio de carpetas que esté claro y se sepa que es lo que está en cada sitio para su mejor organización.

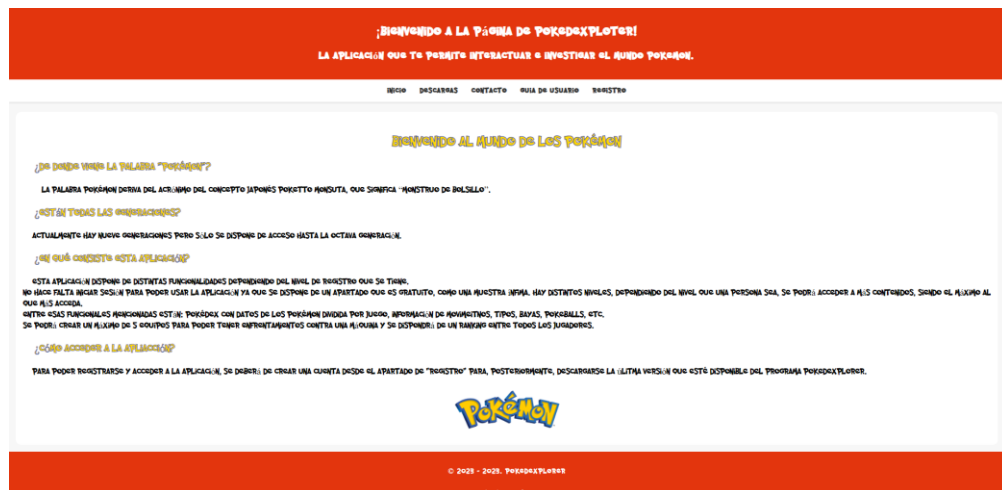


En un futuro, la mejora de usuario se implementará en la página web, con la utilización de la API de *Paypal*, para hacer pagos desde *Paypal* y mejorar así su nivel de usuario.

Explicación del funcionamiento de las distintas páginas disponibles:

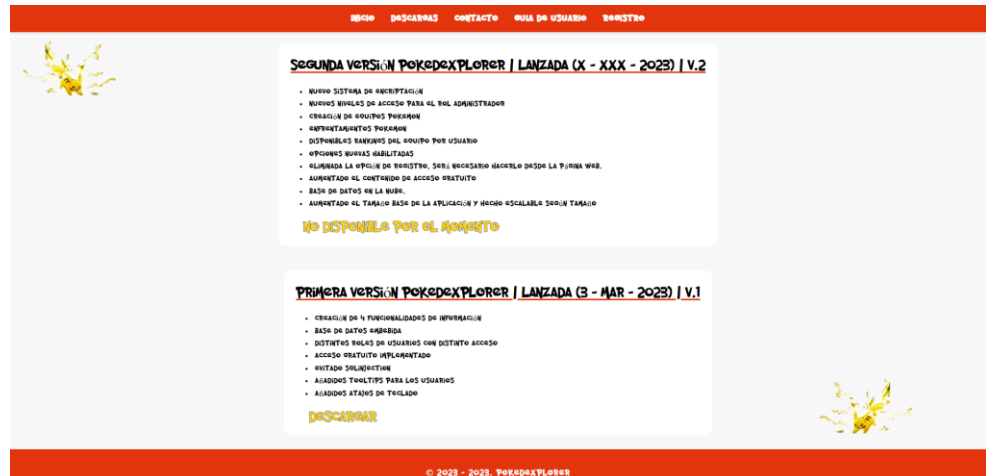
➤ **Página principal.**

En primer lugar, la página que se utiliza con índice al dirigirse a la página web. En esta primera página se habla un poco de en qué consiste y el cómo acceder, además de si están todas las generaciones o de donde provienen lo palabra Pokémon como dato curioso. En un futuro se busca que sirva como un tablón de anuncios, donde salgan noticias en relación con el mundo Pokémon ya sea en su parte de videojuegos o su parte de serie animada.



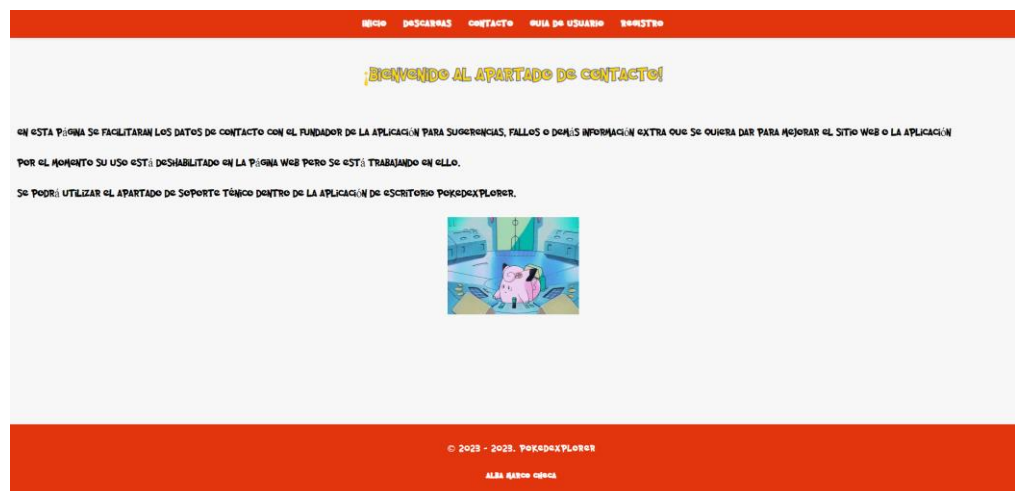
➤ **Descargas**

Esta página se utilizará como un tablón de notas de los cambios llevados a cabo, además de poner a disposición del usuario el instalador de la aplicación.



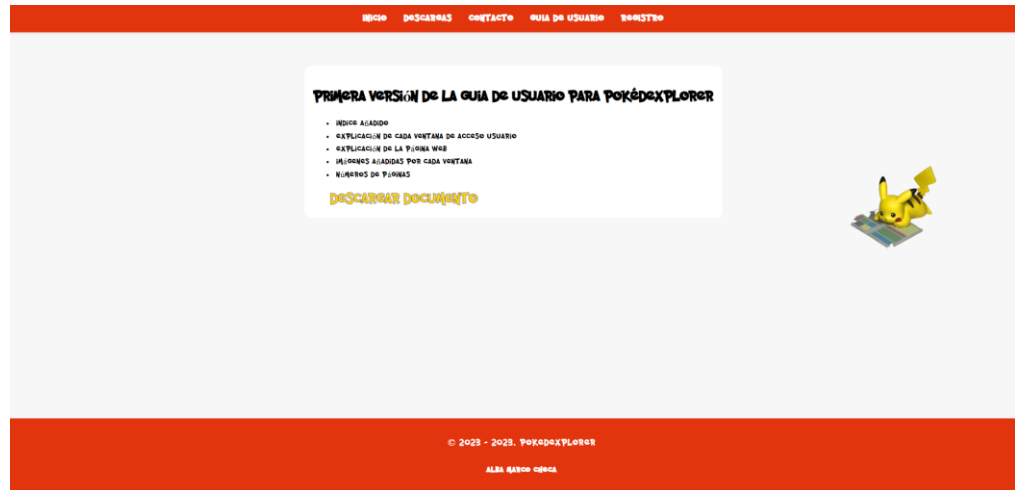
### > Contacto

Por el momento el contacto se hará únicamente mediante el soporte técnico disponible en la página de escritorio. En un futuro se ampliará a su acceso en web, para poder reportar también y/o un correo para ponerse en contacto con administradores, al igual que los tickets.



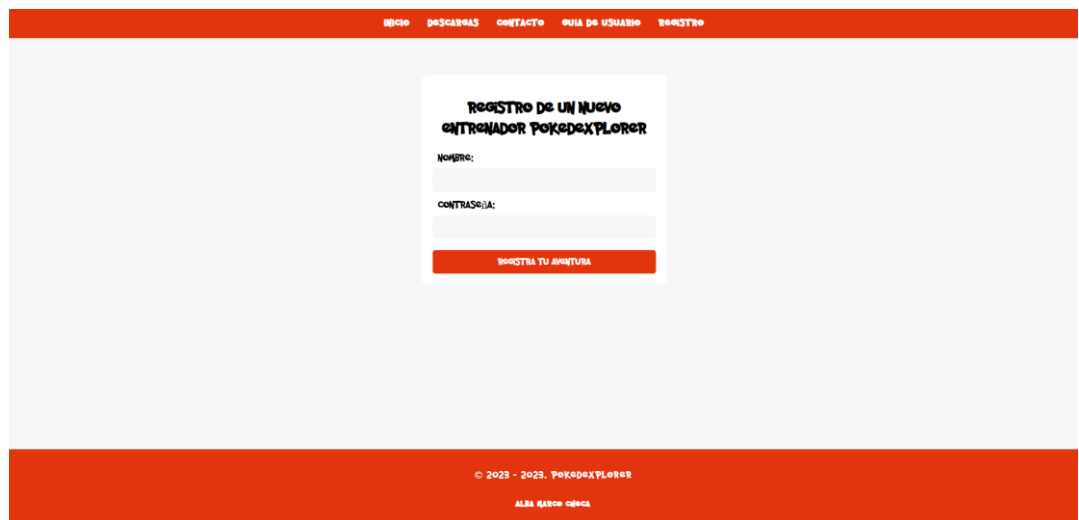
### > Guía de usuario

Se muestra los cambios de la guía de usuario, en donde se explicará el funcionamiento de los diferentes conceptos que se lleva a cabo dentro de la aplicación, en caso de que no se sepa cómo utilizar los diferentes menús.



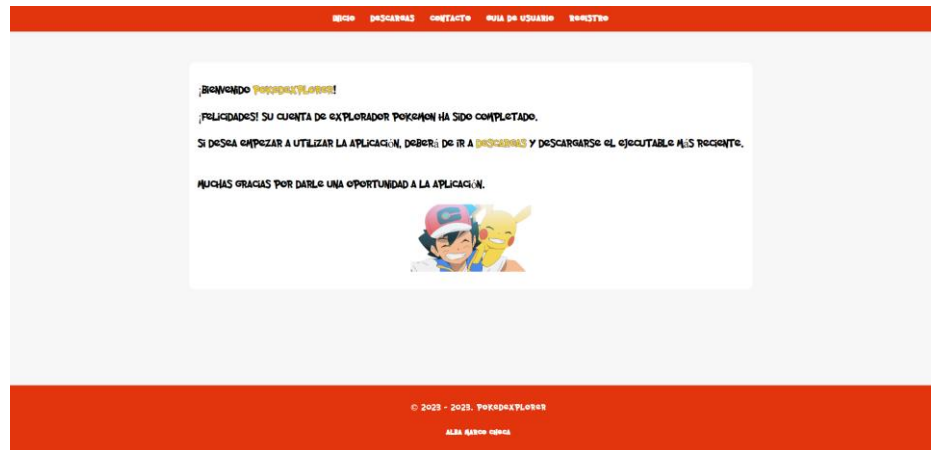
## > Registro

En la ventana de registro, como su propio nombre indica, se lleva a cabo el registro de un usuario, indicando una contraseña y un nombre. En un futuro se pedirán correos electrónicos para, en caso de que acepten, enviarles correos informativos sobre la propia aplicación o mandar los cambios personalmente a cada usuario.

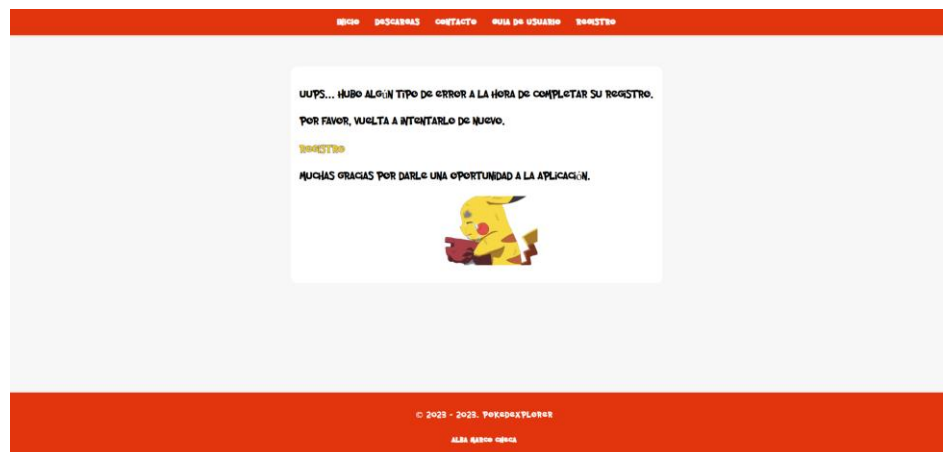


» **Registro correcto**

En caso de que el registro se haya completado correctamente, es decir, si se hizo una conexión a la base de datos y se insertó su campo con su contraseña *hasheada*, se verá su nombre del registro y se le comentará que su registro fue completado correctamente, indicándole que tiene que descargar el instalador.

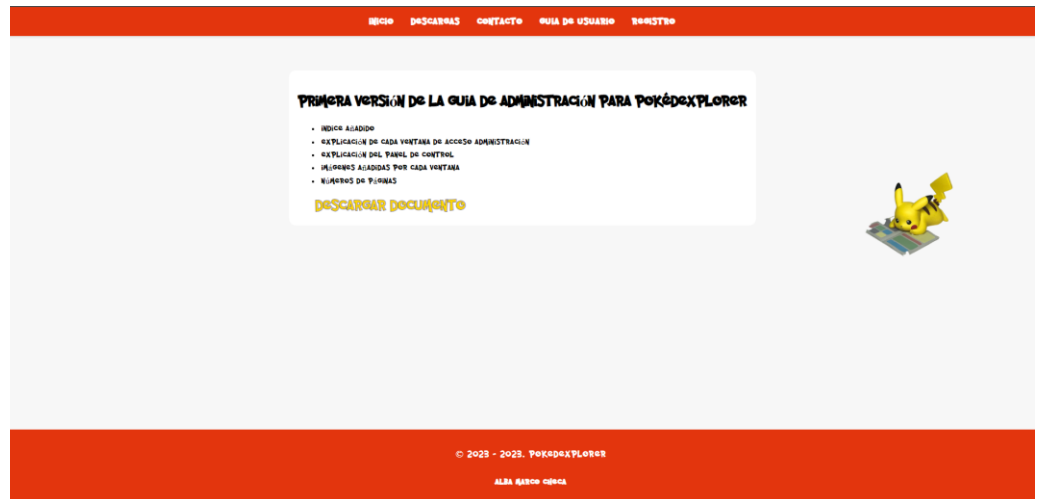
» **Registro incorrecto**

En caso de que esa conexión falle o haya un nombre idéntico, se mostrará el error y la persona se tendrá que volver a registrar.



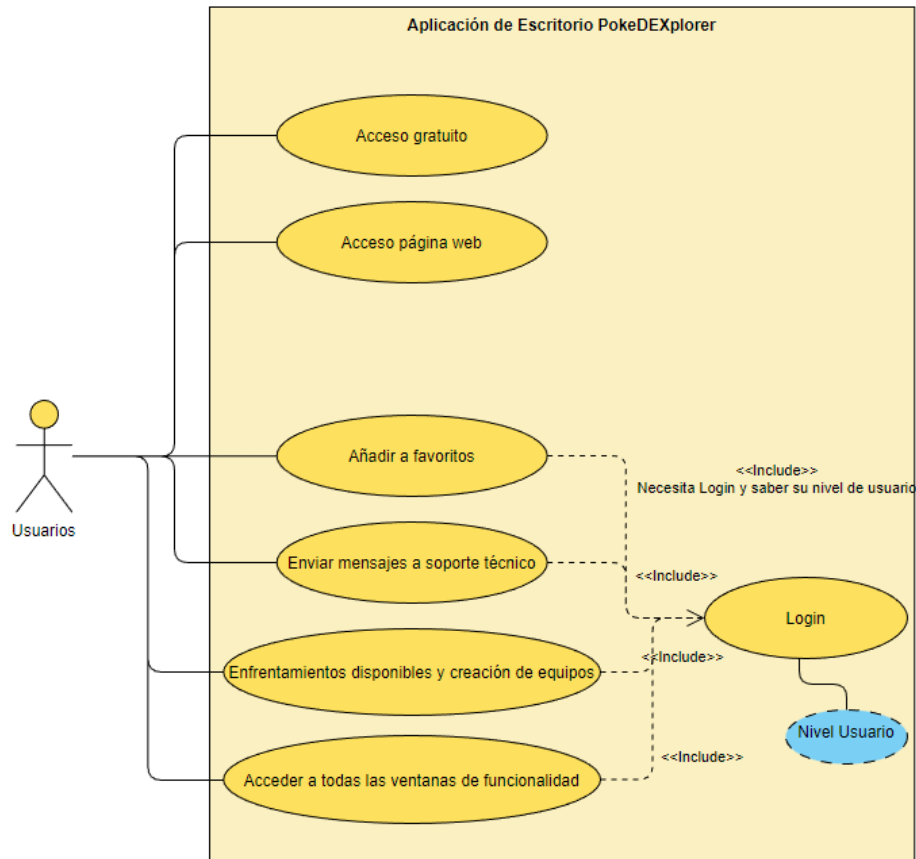
> **Administración** → Guía de administrador.

Esta guía de administrador sólo será accesible desde dentro de la aplicación y desde el menú de administrador, al hacer click en ese menú la redirección vendrá a este enlace, para poder descargar la documentación correspondiente. En esta documentación se explica uno a uno las funcionalidades de cada acción que un administrador puede realizar.



## 6.5. DIAGRAMAS DE USO.

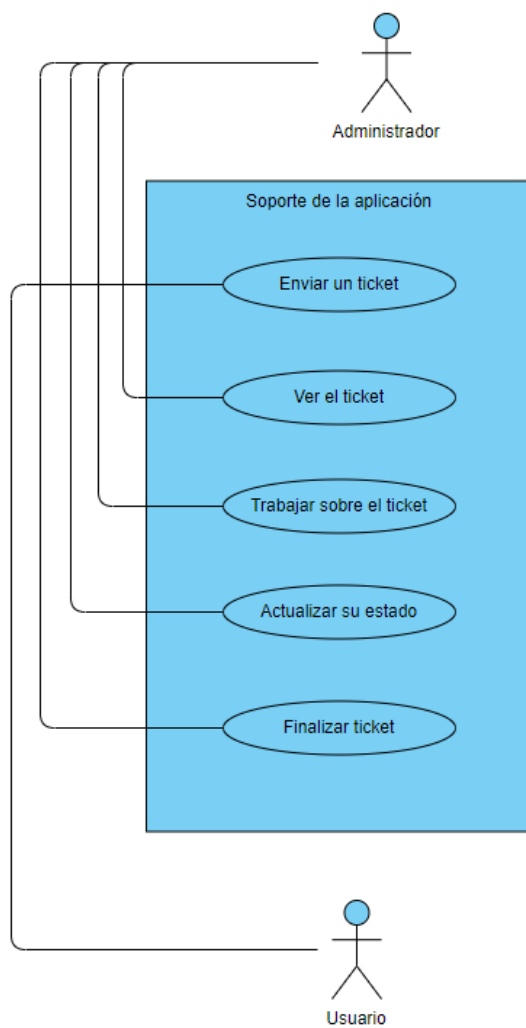
Primer diagrama, sobre el uso de la Aplicación de escritorio **PokeDEXplorer**.



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 [11 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 y el poder añadir a favoritos x elementos junto a su imagen para visualizarlos más tarde. Para acceder a esto, necesitará iniciar sesión y saber su nivel de usuario, de ahí que tenga la relación con “Login” y sea necesario.
- » 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 lo que la página ofrece.

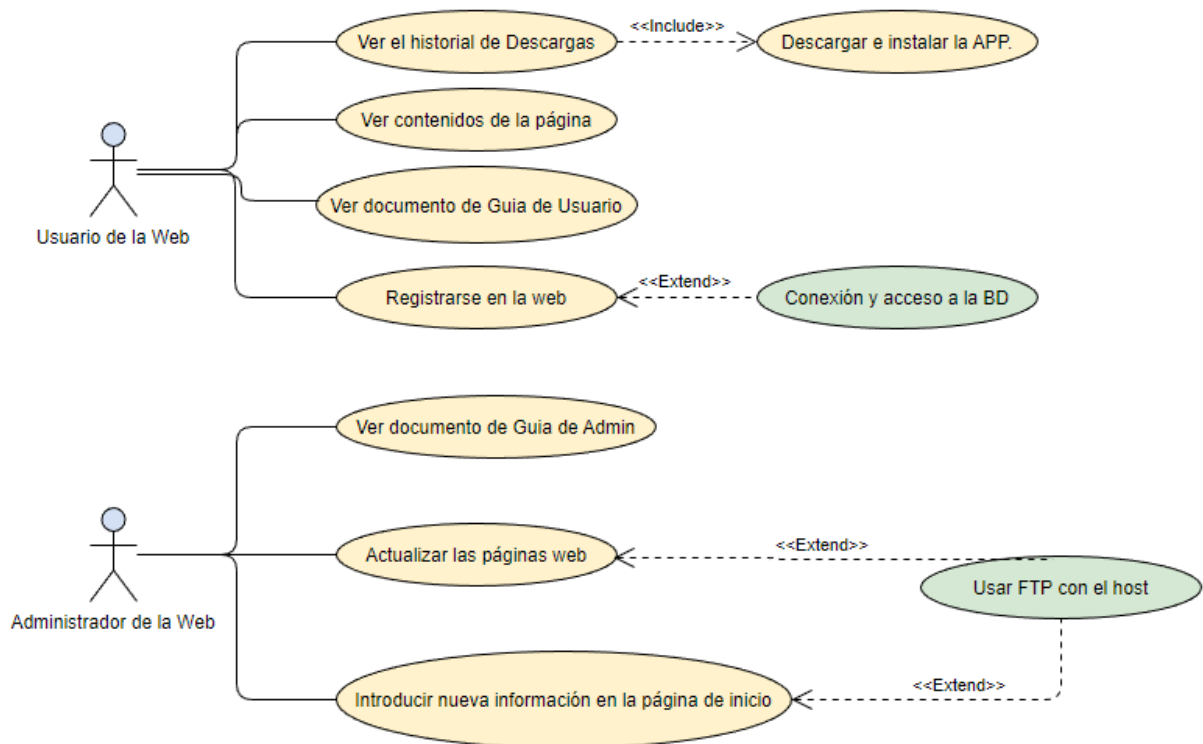




En este diagrama se explica el cómo se va a tratar el tema de soporte técnico / resolución de bugs / explicación de bugs y demás en relación con la aplicación y los usuarios que la utilicen.

- » 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.



- » 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 páginas que están disponibles en el momento y registrarse para usar posteriormente la Aplicación de escritorio. Dispondrá también un acceso a un apartado de documentación para el usuario.
- » 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. Y tendrá a su disposición una guía de administración.

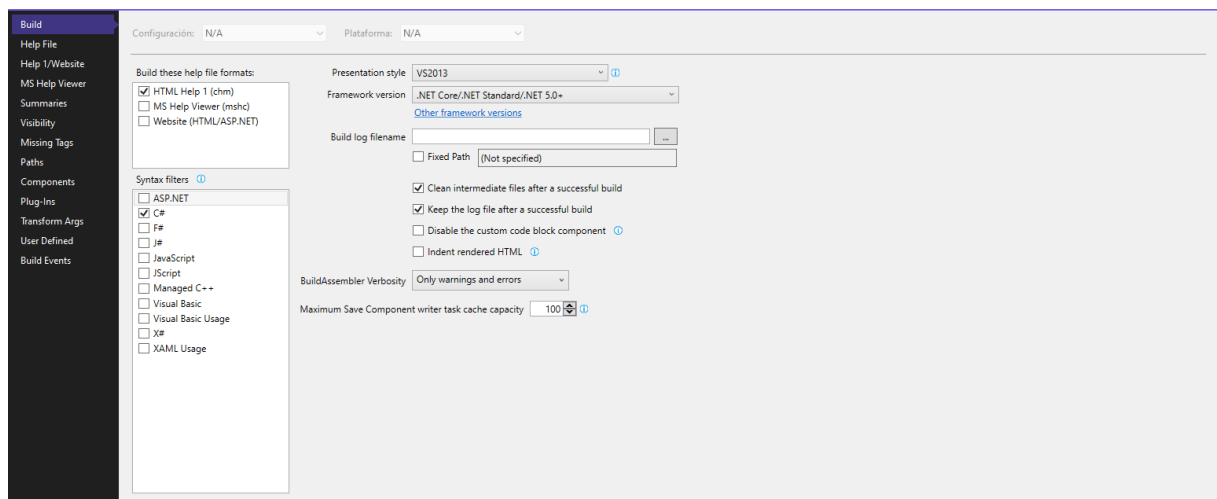
## 7. DESARROLLO.

### 7.1. CREACIÓN, CONFIGURACIÓN E IMPLEMENTACIÓN DE LA BASE DE DATOS.

A la hora de la creación del proyecto de *PokeDEXplorer* se ha de configurar el entorno para una solución, utilizando la plantilla que ofrece Visual Studio 2022 de Aplicación WPF. Su *framework* es .NET 6.0 y, una vez hecho esto, se creará la solución que se necesita junto al proyecto con la plantilla seleccionada.

Al terminar este proceso, se ha de generar un elemento de tipo “*Page.xaml*” para editarlo y crear una página navegable, de tipo *NavigationPage*. Una vez hecho esto, ya estaría creado el proyecto del programa de escritorio, pero se necesitará el proyecto de documentación también.

Para el proyecto de documentación será necesario tener instalado en el equipo **Sandcastle**. Una vez que se tenga instalado, Visual Studio 2022 permitirá crear un proyecto de **SandCastle**, no hay que configurar nada si no se quiere, pero, la herramienta proporciona distintos lenguajes de programación para generar la documentación además de distintos formatos.

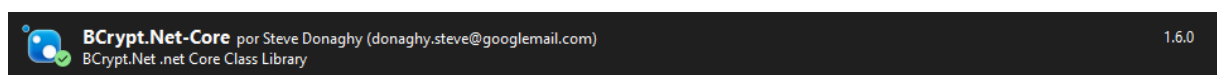


Se deberá de añadir un paquete *NuGet* para añadir el conector necesario para conectar la aplicación a la base de datos que se está utilizando, en este caso de *MySQL*.

Como medida de seguridad a la hora de iniciar sesión, se ha evitado *SQL Injection*, probando y siendo efectivo el método utilizado mediante parámetros en la aplicación de escritorio y mediante comprobaciones de palabras, parámetros y borrado de caracteres raros en PHP.



Aparte de este paquete de NuGet, será necesario el paquete que permite la encriptación y desencriptación que utiliza PHP, para tener el mismo cifrado.



Por último, se ha utilizado la **metodología Cascada** y **SCRUM** para organizar las tareas que se tenían que ir haciendo, añadiéndolas con una descripción para su posterior lectura e implementación de los objetivos que se han propuesto. Según el tiempo y las tareas completadas se van marcando como para testearla y, una vez testeada la tarea se pasa al apartado de completada. De esta manera se lleva una organización más clara del proyecto y de las tareas.

## 7.2. DESARROLLO DE LA INTERFAZ E IMPLEMENTACIÓN DE FUNCIONALIDADES.

En esta etapa se ha trabajado en la creación de una interfaz visualmente atractiva y fácil de usar para el usuario. Para ello, se ha utilizado la herramienta de diseño de WPF, además de la modificación de plantillas y realización de plantillas para los distintos elementos que ofrece la aplicación para hacerlo estéticamente parecido a la temática de Pokémon.

Se han utilizado controles de distintos tipos para que no todo sea similar y de la misma sensación al cambiar de pestaña en las funcionalidades, además de darle a cada una un estilo similar en conjunto pero único en personal.

Se ha buscado en todo momento que para el usuario sea intuitivo todos los controles y su colocación, además de ofrecer la ayuda de *“ToolTip”* (ToolTip C#, s.f.), para así tener una mejor accesibilidad si se deja el cursor en algún elemento, ya que mostrará una pequeña descripción de lo que hará ese componente sobre el ratón.

Para la implementación de las funcionalidades se han creado dos partes de código creadas en bloque, es decir, se ha usado un *“UserControl”* (UserControl C#, s.f.) propio para la aplicación para que ese *“UserControl”* se pueda mover de aplicación en aplicación según su uso y poder reutilizar el código, siendo así más eficiente. Además de la implementación de once funcionalidades más la creación de equipos, en total doce funcionalidades distintas. Esta técnica se ha usado tanto para el menú de administración como el de las funcionalidades de búsqueda y mostrar datos.

Cabe destacar de que en esas funcionalidades se puede agregar a favoritos, por lo que es una implementación que los usuarios pueden tener a su disposición.

### 7.3. PRUEBAS UNITARIAS Y AJUSTES Y MEJORAS.

Durante la creación de las diferentes funcionalidades para su comprobación de correcto funcionamiento, se ha llevado a cabo diferentes pruebas.

La primera prueba, por código y línea a línea, es decir, gracias a la herramienta de Visual Studio 2022 se puede seleccionar bloques de código e ir viendo línea a línea lo que las variables almacenan, lo que hace internamente y demás para poder comprobar que está haciendo la idea principal que se tiene sobre esa funcionalidad.

La segunda prueba es el bloque en conjunto, es decir, todas las funciones desde el programa principal para ver cómo funciona, si los datos mostrados son correctos o si no lo son.

Se han realizado ajustes según se iba comprobando el código, línea a línea, como mejoras de código, limpieza de código, comprobación de la integridad de los datos, gestión de excepciones y/o errores de código para su correcto funcionamiento y un rendimiento mejorado, sin que la aplicación tenga que parar y otorgando así una mala experiencia al usuario.

### 7.4. DESARROLLO DE PERFILES DE USUARIO / 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 y a visualizar la página web.

- › **Nivel básico**, en base de datos estará representado como nivel “1”. En este nivel se tendrá acceso a lo anterior, más acceso al soporte técnico, más dos ventanas de funcionalidades, como sería ver Pokémon buscando por nombre o ID, 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 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.

### 7.5. DESARROLLO DE CONCURRENCIA EN LA APLICACIÓN.

El desarrollo de concurrencia (Concurrencia C#, s.f.) 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 secundario 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.

Además de lo comentado, para que no pulsen el botón antes de que se termine de cargar la lista ya si lo pulsan generará un error por la constante actualización, se utiliza una barra de progreso con un máximo según registros haya en la consulta que se esté llevando a cabo, llenando así la lista.

- » Al entrar al apartado de enfrentamientos. La idea es hacer un hilo principal que maneje el tiempo 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 qué 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.

## 7.6. DESARROLLO DE SEGURIDAD EN LA APLICACIÓN.

Se ha tenido en cuenta durante la creación de todo el proyecto la seguridad que este tiene que otorgar a un usuario y a la integridad de sus datos y por ende la confiabilidad los datos a la hora de ingresarlos en base de datos.

- › Autenticación y accesos → Para este apartado se ha querido implementar la validación de sus credenciales y la asignación de roles a los distintos usuarios y distintos administradores con accesos diferentes.
- › Protección de datos → Todos los datos más vulnerables o sensibles se han implementado mediante un sistema de encriptación, en este caso **CRYPT\_BLOWFISH** para así asegurar lo que se quiere proteger.
- › Prevención de ataques → En todos los campos que se puedan introducir datos y tengan acceso a la base de datos y/o interna de la aplicación se han desarrollado validando su entrada de datos y protección contra inyección de código como sería el famoso **SQL Inejction**, tanto en parte web como en web escritorio.
- › Gestión de errores → Gestión de errores en todos los apartados de la aplicación, para evitar su cierre y así no perder los datos que se están usando y tenerlos disponibles para asegurar la integridad de los mismos.
- › Ofuscación del código en la parte escritorio → En la parte web, al ser parte de servidor no trabaja con ofuscación, se realizó una validación de entrada tanto en HTML como en PHP para asegurar los datos. Para la parte de la aplicación se ha utilizado Babel Ofuscator (Babel Ofuscator, s.f.), un ejecutable fácil de usar y compatible con la versión del proyecto. Se ha completado la descarga desde su página gratuita para posteriormente instalarlo en el proyecto y así conseguir, con su ejecutable DEMO una aplicación ofuscada, mejorando su seguridad al ser un ejecutable.

### Evento de posterior a la compilación

Especifica los comandos que se ejecutan después de completar la compilación. No se ejecuta si la compilación falla. Utilice "llamar" para invocar archivos .bat. Un código de salida distinto de cero hará que falle la compilación.

```
"D:\BabelOfuscator\babel.exe" "${TargetPath}"
```



```
1>AppDI -> C:\Users\almar\Desktop\Proyectos\FinalDAM\AppDI\AppDI\bin\Debug\net6.0-windows\AppDI.dll
1>Babel Obfuscator Demo version 9.9.5.0
1>Copyright (C) 2010-2022 babelfor.NET. All rights reserved.
1>Processing C:\Users\almar\Desktop\Proyectos\FinalDAM\AppDI\AppDI\bin\Debug\net6.0-windows\AppDI.dll...
1>Analyzing AppDI.dll...
1>Target framework: .NETCoreApp,Version=v6.0
1>Running agent...
1>Control flow obfuscation...
1>Obfuscating names...
1>Encrypting strings...
1>Writing C:\Users\almar\Desktop\Proyectos\FinalDAM\AppDI\AppDI\bin\Debug\net6.0-windows\BabelOut\AppDI.dll
1>EXEC : warning [W000000]: This is an evaluation version, the obfuscated assembly will no longer work after 3
1>Writing C:\Users\almar\Desktop\Proyectos\FinalDAM\AppDI\AppDI\bin\Debug\net6.0-windows\BabelOut\AppDI.xml
1>
```

- › Revisiones de seguridad → Una vez lanzada la aplicación al mercado, se irán mejorando los sistemas de seguridad para identificar posibles problemas y mejorar la seguridad del sistema.

## 8. DESARROLLO EMPRESARIAL.

### 8.1. PRESENTACIÓN DEL PROYECTO Y DEL EMPRENDEDOR.

#### a. PRESENTACIÓN BREVE DEL PROYECTO.

*PokeDEXplorer* es una aplicación que ofrece información sobre Pokémon, con diferentes niveles de usuarios, desde el gratuito hasta otros cuatro que ofrecen más beneficios. Los usuarios pueden agregar información a favoritos, reportar errores y sugerencias a través de tickets de soporte y crear equipos para poder hacer enfrentamientos. Está enfocado sobre todo en la búsqueda de la información.

#### b. INTERÉS Y CAPACIDAD EMPRENDEDORA.

Se inicia este proyecto porque a la desarrolladora le apasiona el mundo de los videojuegos y por ende la creación de una aplicación referente a un juego que ha conseguido gran base de jugadores. Se interesa por este proyecto debido a que cree que hay una gran demanda de información sobre Pokémon y, la que hay actualmente, está de forma poco intuitiva para las personas que la buscan. Busca mejorar el juego de los jugadores con una interfaz fácil de usar y que resulte familiar con la estética de Pokémon, además de poner toda la información en un único lugar, sin tener que buscar en varios sitios.

## 8.2. IDEA DE NEGOCIO.

### a. REDACCIÓN DE LA IDEA Y PROPUESTA DE VALOR Y OBJETIVOS.

La idea de negocio es crear una aplicación de escritorio llamada *PokeDEXplorer* que proporcione a los usuarios diferentes niveles de acceso a funciones y datos. La aplicación se centrará en proporcionar información en primer lugar, añadiendo más información en un futuro, además de herramientas de gestión, como la posibilidad de agregar elementos a favoritos, reportar errores y sugerencias a través de tickets de soporte, y crear equipos de seis integrantes con un máximo de cinco equipos para, en un futuro, implementar la funcionalidad de hacer enfrentamientos.

La propuesta de valor de *PokeDEXplorer* es ofrecer a los usuarios una aplicación de escritorio de alta calidad, intuitiva y familiar que les permita a los usuarios acceder a aquella información que está en distintos sitios en un solo sitio de información y herramientas útiles.

El objetivo principal es proporcionar una experiencia de usuario satisfactoria y mejorar constantemente la aplicación a medida que la base de usuarios crece.

## 8.3. MERCADO Y CLIENTES.

### a. TIPO DE MERCADO.

El mercado objetivo de *PokeDEXplorer* es el de las aplicaciones de escritorio. Es un mercado en crecimiento que se espera que siga creciendo en el futuro. La aplicación se centrará principalmente en usuarios que buscan información sobre el videojuego Pokémon, estando todo en un mismo lugar y, alguna herramienta de gestión para marcar algo en favorito y tenerlo más visible. Posteriormente se buscará en una adaptación para el teléfono móvil, para que esté más accesible la información para aquellas personas que no tengan un ordenador. Y, para el caso de que no tengan teléfono móvil propio, se buscará en un futuro implementarse en la página web que se ha creado, con un inicio de sesión al igual que en la aplicación de escritorio.

### b. SEGMENTO DE MERCADO.

El segmento de mercado objetivo de *PokeDEXplorer*, de primeras, son los usuarios de aplicaciones de escritorio que buscan información sobre este videojuego. El segmento se divide en dos subgrupos: los usuarios gratuitos y los usuarios que pagan por acceso a funciones adicionales.

**c. ESTUDIO DE MERCADO: LOS CLIENTES.**

La investigación de mercado realizada indica que hay una demanda insatisfecha de aplicaciones de escritorio que ofrezcan información en un solo lugar. Los usuarios estarían más interesados en un lugar donde esa información a la que acceden estén toda en un solo lugar, de manera conveniente y fácil de usar. Hay que tener en cuenta los rangos de edades, al ser un videojuego puede jugarlo gente menor de edad o gente mayor de edad, por ende, en la aplicación se deberá de utilizar vocabulario adecuado para todos los rangos de edades y que sea entendible para todos.

**8.4. ENTORNO Y COMPETENCIA.****a. ENTORNO GENERAL Y ESPECÍFICO.**

La industria de las aplicaciones informáticas está en constante crecimiento y evolución, por lo que se trata de un entorno muy dinámico y activo. La creciente popularidad de las aplicaciones móviles y el aumento de la conectividad a internet en todo el mundo han creado un entorno favorable para el crecimiento de aplicaciones de este tipo y más con el objetivo de ampliar a todo tipo de plataformas para su amplio uso y adecuarse a las necesidades del usuario.

*PokeDEXplorer* se encuadra en el sector de las aplicaciones de información y entretenimiento para el usuario. Su público objetivo son usuarios de escritorio que buscan información sobre los videojuegos de Pokémon para agregar dicha información a favoritos y tener una mejor accesibilidad que en una página cualquiera sin registro y sin favoritos.

En un futuro se buscará ampliar lo que ofrece, permitiendo realizar combates tanto en contra de la IA y en contra de los usuarios registrados.

**b. COMPETENCIA.**

La aplicación compite en un mercado en el que existen varias páginas web similares que ofrecen información y la posibilidad de competir en línea. Algunos de los competidores directos son *Pokemon ShowDown* que permite tener combates en línea con distintas categorías según lo que el usuario seleccione, o aplicaciones como *PokeMMO* que es el juego completo en línea, donde los usuarios se encontrarán con otros usuarios como si fuera el videojuego real. Estas aplicaciones tienen una base de usuarios sólida y han logrado establecerse en el mercado. Y conseguir su propia fama entre los distintos usuarios.

También hay que tener en cuenta las páginas de información, que serán las que primero saldrán según su posicionamiento en el motor de búsqueda que se utilice, es decir, ya tienen su posición ganada en el buscador y saldrá antes como resultado que la propia aplicación *PokeDEXplorer*.

Se ha de desarrollar una estrategia efectiva de diferenciación para destacar entre sus competidores y atraer nuevos usuarios. Como podría ser utilizar SEO para redirigir a los motores a la página web para que la gente se registre y descarguen la aplicación para su utilización.

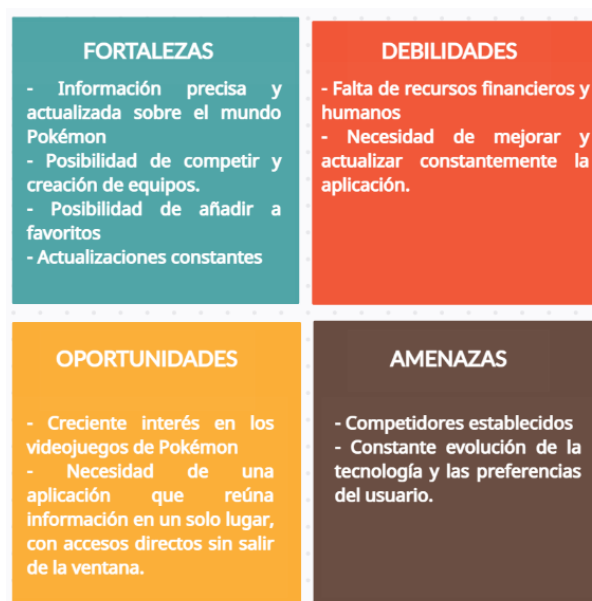
### c. ANÁLISIS DAFO.

Debilidades: Falta de recursos financieros y humanos para el momento actual, y la necesidad de mejorar y actualizar constantemente la aplicación.

Amenazas: Competidores establecidos y la constante evolución de la tecnología y las preferencias del usuario.

Fortalezas: Ofrece información precisa y actualizada sobre el mundo Pokémon, posibilidad de competir y creación de equipos, y un enfoque en la comunidad de jugadores, posibilidad de añadir a favoritos, actualizaciones constantes.

Oportunidades: El creciente interés en los videojuegos de Pokémon y la necesidad de una aplicación que reúna información en un solo lugar, con accesos directos sin salir de la ventana.



**d. LOCALIZACIÓN DE LA EMPRESA.**

La localización de la empresa, al ser una empresa pensada para trabajar una sola persona, no tendrá un lugar determinado donde estar la empresa ya que es indiferente desde donde se trabaje. En caso de crecimiento de la aplicación y de las necesidades de comenzar la contratación de más gente, se mirará un lugar el cuál esté bien comunicado por carretera, con una conexión cableada que sea rápida para todo el equipo que esté trabajando. Todo esto siempre y cuando el presupuesto para poder alquilar el lugar no se vaya fuera del alcance que se podrá dar con el dinero que *PokeDEXplorer* vaya generando.

**e. MISIÓN, VISIÓN Y VALORES.**

La misión final de *PokeDEXplorer* es convertirse en la aplicación de referencia para los jugadores de Pokémon, ofreciendo información precisa y actualizada sobre todos los ámbitos que ofrece el mismo juego y un lugar donde albergar los favoritos, así como en el futuro la posibilidad de tener un apartado para poder competir y crear equipos.

La visión que se espera es ser una empresa, al principio pequeña que irá en crecimiento según usuarios tengamos, líder en el sector de las aplicaciones de información y entretenimiento para los usuarios en cuanto a Pokémon se refiere. Ofreciendo una aplicación innovadora, ágil, familiar y con fácil acceso. Consiguiendo que la experiencia hacia el usuario sea más que satisfactoria.

Los valores de la empresa son la calidad de la aplicación entregada con actualizaciones y mejoras constantes, la innovación con la actualización de nuevos métodos y nuevas funcionalidades, la transparencia hacia los usuarios indicándoles en qué se está trabajando y en que se trabajará cuando se muestre la actualización de aplicación y el compromiso con la comunidad de jugadores del mundo Pokémon para ofrecerles una aplicación donde todo aquello que está en diferentes sitios esté en un solo lugar. Se busca dar sencillez con para mejorar la experiencia de la interfaz, siendo cómoda y ágil.

**f. RESPONSABILIDAD SOCIAL CORPORATIVA.**

*PokeDEXplorer* se compromete a utilizar prácticas éticas y responsables en todas sus operaciones.

Se asegurará de que sus empleados sean tratados con respeto y dignidad, y promoverá la igualdad de oportunidades y la diversidad en su fuerza laboral. También se compromete a

respetar el medio ambiente con las energías renovable y a trabajar de manera sostenible y responsable.

Todo lo mencionado anteriormente tendrá una responsabilidad económica para que sean consecuentes y coherente económicamente hablando.

## **8.5. ESTRATEGIAS DE MARKETING.**

### **a. ESTRATEGIA DE POSICIONAMIENTO, PRODUCTO, PRECIO, DISTRIBUCIÓN.**

La estrategia de posicionamiento se basará en destacar la oferta de valor de la aplicación, enfocándose en la calidad y utilidad de la información proporcionada ya sea buscándola de forma general o en los favoritos, así como en la posibilidad de crear equipos y enfrentamientos entre ellos. Se utilizarán canales de comunicación digitales, como las redes sociales o foros de Pokémon y la publicidad en línea junto a un posicionamiento SEO para su página web, para llegar al público objetivo y al público que esté interesado en este proyecto.

La estrategia del desarrollo producto se enfocará en la evolución constante de la aplicación, con la implementación de nuevas funcionalidades y mejoras en la experiencia del usuario. Se ofrecerá un nivel gratuito para atraer a los usuarios y se promocionarán las funcionalidades adicionales de los niveles superiores. Ya sea con una prueba gratuita periódica o mostrándolo en la página web.

La estrategia de precios se basará en ofrecer un nivel gratuito y varios niveles de pago para satisfacer las necesidades y presupuestos de los distintos segmentos de mercado. Se llevará a cabo un análisis de precios de la competencia y se ofrecerán precios competitivos en el mercado. Se ofrecerán precios bajos en general debido a que toda esta información está de forma gratuita en todo internet, lo que hace la aplicación es facilitar toda la información en un mismo lugar.

La distribución se realizará a través de descargas en línea, utilizando los principales canales de distribución de software para llegar al público objetivo como sería desde la página propia de *PokeDEXplorer*. Además, se promocionará la aplicación a través de la publicidad en línea, posicionamiento SEO y las redes sociales o foros en relación con la aplicación.

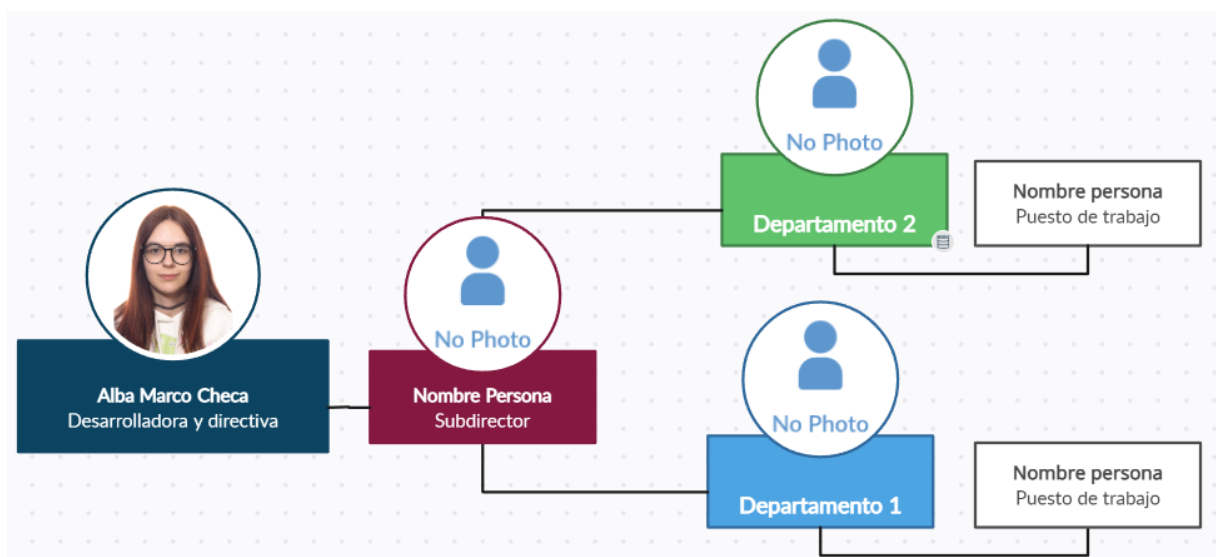
**b. ATENCIÓN AL CLIENTE.**

Se establecerá un servicio de atención al cliente, primeramente, en la aplicación de escritorio para posteriormente agregar la funcionalidad en soporte móvil y en la propia página web, para recibir y resolver las preguntas, dudas, errores y sugerencias que puedan tener los usuarios. Además, se fomentará la participación de los usuarios para mejorar la calidad de la aplicación a través de encuestas y formularios periódicamente. Se podrá acceder sin tener una cuenta registrada, para facilitar su uso y su acceso para todo tipo de usuarios, incluso para los gratuitos.

**8.6. RECURSOS HUMANOS.****a. ORGANIGRAMA DE LA EMPRESA.**

La empresa se organizará en torno al fundador, quien será responsable de la gestión y coordinación de los distintos departamentos y recursos de la empresa, como el equipo de desarrollo, atención al cliente, finanzas, marketing o futuros departamentos que se vayan necesitando según se avance el crecimiento de la aplicación.

Por el momento el fundador, que sería el desarrollador de la aplicación *PokeDEXplorer* será quien lleve todas las funcionalidades de los diferentes departamentos mientras que la aplicación se expande.

**b. DISEÑO PUESTOS Y SELECCIÓN DE PERSONAL.**

Al principio, no se llevará a cabo contratación ya que no habrá suficiente dinero al estar empezando para tener personal y que tengan un sueldo mínimo. Una vez que la aplicación

obtenga fama y los distintos niveles de usuarios van creciendo, se considerará la posibilidad de que haya una contratación según necesidades. La selección de personal se haría según las necesidades de la aplicación, es decir, si es necesario tener más personal con conocimientos en C# se buscará ese perfil y se harán pruebas simples para validar unos conocimientos mínimos.

**c. MOTIVACIÓN LABORAL.**

Se establecerán planes de motivación laboral para fomentar la productividad y satisfacción de los empleados. Se ofrecerán salarios competitivos, beneficios y un ambiente de trabajo cómodo y seguro. Además, se podrá tener la posibilidad de teletrabajar, lo que puede ser más cómodo y más llamativo que puestos desde una oficina. Se acordarán horarios flexibles, los cuales se adapten al trabajador para que su productividad aumente. Se darán todas las herramientas necesarias para desarrollar su puesto de trabajo, como equipos de sobremesa en caso presencial o equipos portátiles para el teletrabajo. Si se desea una jornada híbrida se le otorgará portátiles para teletrabajar y venir a oficina con su propio ordenador personal.

**d. CONTRATACIÓN Y COSTE TRABAJADORES.**

La contratación y el coste de los trabajadores variará según las necesidades de la aplicación, es decir, si la aplicación no está triunfando, no se tendrá del capital suficiente para poder contratar y sostener a los trabajadores que haya en los departamentos. Según tiempo trabajado en la empresa se podrá mirar un aumento fijo que acompañe al sueldo, aparte de los aumentos salariales según productividad y desarrollo como trabajador en la propia empresa. Se buscará contratación en plataformas online, con entrevista física para llevar a cabo una prueba de conocimientos sobre lo que postuló la persona en cuestión.

**e. ORGANIZACIÓN DE LA PREVENCIÓN.**

Se establecerán medidas de prevención y seguridad laboral para garantizar un ambiente de trabajo seguro y saludable.

Entre estas medidas de prevención están el cuidado de la postura del trabajador, ofreciendo una sillas de oficina cómodas y según necesidades para ella, descanso de jornada para la fatiga visual, mesas elevables para la fatiga muscular al estar sentado, una iluminación correcta por habitación, diseño de instalaciones con reguladores de temperatura para todo tipo de climas,



protección para los golpes en muebles y/o paredes, disponibilidad de agendas para los trabajadores, minimizar el ruido de las oficinas para una mayor concentración, etc.

### **8.7. FORMA JURÍDICA.**

#### **a. FORMA JURÍDICA Y MOTIVOS.**

Se ha elegido la forma jurídica de una empresa unipersonal para comenzar debido a su simplicidad y bajo costo de registro. Además, proporciona al fundador el control total de la empresa. Como ya se ha explicado con anterioridad, no se tiene la idea de empezar a contratar personal en la primera fase de crecimiento de la aplicación. Es decir, si la aplicación no tiene éxito no se podrá avanzar a otros niveles de formas jurídicas como sería Sociedad Civil o Sociedad Limitada que serían los siguientes niveles para poder crecer poco a poco y seguir aumentando el capital para aumentar el personal y las instalaciones. (Formas Jurídicas, s.f.)

### **8.8. PLAN DE PRODUCCIÓN.**

#### **a. PLAN PRODUCCIÓN Y PROVEEDORES | GESTIÓN DE ALMACÉN E INVENTARIOS.**

El plan de producción no se ve necesario ya que no es una empresa sobre productos de compra venta sino de desarrollo de software. Como mucho se puede realizar un plan de desarrollado, es decir, una gestión sobre el tiempo para llegar a X objetivos que aplique el desarrollador.

Estos planes irán según trabajadores contratados y según trabajo se tenga, es decir, no siempre se va a estar con la misma cantidad de trabajo y los plazos irán variando según esta carga.

### **8.9. ANÁLISIS ECONÓMICO - FINANCIERO.**

#### **a. ANÁLISIS DE COSTES [FIJOS, VARIABLES, UMBRAL RENTABILIDAD Y PRECIO DE VENTA].**

Actualmente, no hay ningún tipo de coste fijo ni variable ya que se están utilizados hostings gratuitos tanto para la página web como para la base de datos.

Según haya un crecimiento de la aplicación, progresivamente se irán aumentando estos hostings con una mejor prestación para darle a los usuarios una comodidad y/o rapidez mayor. Mediante redes sociales tampoco serán costes debido a que son gratuitas y subir contenido es gratuito totalmente, además de poder llegar a tener pago de ellas.

El umbral de rentabilidad por el inicio de la empresa sería positivo en todo momento debido a que se están usando, como se comentó, todo de forma gratuita y no hay costes provenientes de ningún lugar. Según la empresa crezca, se irán aumentando los costes y el umbral según suscripciones.

**b. INVERSIONES Y GASTOS INICIALES.**

Únicamente se haría una pequeña inversión en publicidad o en herramientas SEO para un mejor posicionamiento de la página web.

Se podría también hacer uso de redes sociales que sería todo gratuito. Aunque se puede invertir en anuncios para un umbral de personas interesadas en Pokémon.

**c. PLAN FINANCIACIÓN.**

En primer momento no será necesario ya que se prevé utilizar hostings que sean gratuitos y, según haya suscripciones se irá viendo la posibilidad de diferentes opciones de pago tanto para el hosteo de la página web como el hosteo de la base de datos.

En un futuro incluso se puede invertir en una máquina propia, en una sala especializada para ella donde se lleve a cabo la instalación completa de los servicios necesarios, así como de programas que sean necesarios para el escritorio remoto o uso para otro tipo de aplicaciones.

**d. PREVISIÓN DE RESULTADO Y BALANCE DE SITUACIÓN.**

A corto plazo no será problema la situación ya que no se espera que sean valores negativos de capital al usar herramientas totalmente gratuitas tanto de hosting como de desarrollo y pruebas.

A largo plazo, sí que será algo que hay que ver con calma junto a un equipo de finanzas para llevar a cabo un plan financiero acorde a la situación en que la aplicación esté en ese momento.

Según el crecimiento que se tenga se irá invirtiendo más capital hasta obtener herramientas más competitivas en el sector para ofrecer al usuario un mayor comodidad, seguridad y rapidez a la hora de usar los servicios que se ofrecen con *PokeDEXplorer*.

**e. PLAN TESORERÍA.**

El plan de tesorería que se espera es que mensualmente se vayan aumentando gradualmente durante el tiempo de vida de la aplicación. Se espera también que tampoco

haya deudas con nadie y, los gastos nunca superen a los ingresos. La empresa deberá de ser capaz de mantener sus operaciones y las inversiones para generar más capital.

**f. ESTIMACIÓN DE IMPUESTOS.**

Como mínimo se tendrá en cuenta el impuesto de Sociedades, Seguridad Social, Impuesto sobre la renta y, en caso de empleados, su Seguridad Social según precios aplicables en España.

También, se tendrá en cuenta el 21% de IVA que se aplicará al comprar el nivel distinto de usuario y los porcentajes de las distintas plataformas y/o aplicaciones que se tiene previsto obtener para así mejorar la propia aplicación.

**8.10. PLAN DE PUESTA EN MARCHA.**

**a. TRÁMITES Y PLAZOS Y DOCUMENTACIÓN ADMINISTRATIVA - CONTABLE.**

- › Registro de la empresa:
  - » Obtener el NIF/NIE.
  - » Registro de la empresa en el Registro Mercantil.
  - » Inscripción en la Seguridad Social y en la Agencia Tributaria.
- › Diseño y desarrollo de la aplicación:
  - » Registrar la marca de la aplicación y el nombre.
  - » Definir los requisitos de la aplicación.
  - » Diseñar la arquitectura y la interfaz gráfica de la aplicación.
  - » Desarrollar la aplicación con la herramienta de desarrollo elegida.
  - » Realizar pruebas para garantizar el correcto funcionamiento de la aplicación.
- › Marketing y publicidad:
  - » Diseñar y desarrollar la estrategia de marketing.
  - » Crear perfiles en redes sociales.
  - » Realizar campañas publicitarias en línea y offline.
- › Lanzamiento de la aplicación:
  - » Publicar la aplicación en la plataforma elegida.
  - » Anunciar el lanzamiento de la aplicación en la página web y redes sociales.
  - » Preparar una estrategia de atención al cliente para el lanzamiento.
- › Monitoreo y mejora continua:

- » Monitorear el rendimiento de la aplicación y la satisfacción del cliente.
- » Identificar áreas de mejora y realizar mejoras de forma continua.
- » Realizar actualizaciones y correcciones necesarias en la aplicación.

## 9. Conclusiones.

En conclusión, sobre los objetivos que se han querido realizar en este proyecto, se han cumplido satisfactoriamente, cumpliendo todos los requisitos previamente definidos de cada objetivo, es decir, el trabajo realizado satisface las necesidades con un futuro a desarrollar, con más objetivos y más funcionalidades que no se han implementado como objetivo para este proyecto.

Se ha buscado en todo momento que la usabilidad y funcionalidades de la aplicación implementadas es efectivas y fáciles de usar para los usuarios gracias a ToolTips sobre cada elemento de la aplicación, ayudando así con una explicación del componente. Se busca mejorar en el aspecto de la accesibilidad poniendo a disposición del usuario teclas de atajo.

Respecto al rendimiento de la aplicación, se deberá de mejorar el acceso a base de datos con el propio host, ya que la respuesta, en general, suele ser lenta y un poco tedioso a la hora de iniciar sesión, respecto las acciones de administradores y demás también se nota un poco de lentitud a la hora de realizar consultas. Las consultas a API bastantes amplias de registro también se demoran un poco, pero eso es problema de la propia API de donde se sacan los datos y el tipo de conexión que se lleva a cabo.

El problema en específico que se comentó con anterioridad se ha solucionado, juntando toda esa información en una misma pantalla, separada únicamente por pestañas dentro de la ventana, mejorando eficazmente tal y como está actualmente el manejo de información en las diferentes webs que ofrecen esta información.

Una de las limitaciones que se han visto durante la realización es la base de datos, ya que al ser un hosteo un poco lento la información la procesaba más lento. Se busca en versiones futuras mejorar ese mejor hosteo para llevar a cabo las acciones más rápido. También se busca la implementación de la funcionalidad de los combates Pokémon y la mejora

de la adicción a favoritos con más campos, para su posterior vista en la ventana correspondiente.

Se espera que el impacto potencial de la aplicación a corto plazo sea ínfimo debido a su poca fama entre los usuarios hasta que vaya creciendo su fama poco a poco. Además, se busca su mejora gracias al apartado de sugerencias y/o objetivos que se planean como futuros usando la metodología SCRUM.

## 10. Bibliografía.

*Babel Ofuscator.* (s.f.). Obtenido de <https://www.babelfor.net/>

*Concurrencia C#.* (s.f.). Obtenido de <https://csaldias.dev/posts/csharp-concurrency/>

*Descarga de fuentes.* (s.f.). Obtenido de <https://fonts2u.com/>

*Diseño Responsivo.* (s.f.). Obtenido de <https://www.freecodecamp.org/espanol/news/diseño-web-responsive-como-hacer-que-un-sitio-web-se-vea-bien-en-telefonos-y-tabletas/>

*Documentación MySql y .NET.* (s.f.). Obtenido de <https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-intro.html>

*Documentación NavigationWindow.* (s.f.). Obtenido de <https://learn.microsoft.com/en-us/dotnet/api/system.windows.navigation.navigationwindow?view=windowsdesktop-7.0>

*Documentación SandCastle.* (s.f.). Obtenido de <https://ewsoftware.github.io/SHFB/html/bd1ddb51-1c4f-434f-bb1a-ce2135d3a909.htm>

*Formas Jurídicas.* (s.f.). Obtenido de <https://plataformapyme.es/es-es/IdeaDeNegocio/Paginas/FormasJuridicas.aspx>

*Hash C#.* (s.f.). Obtenido de <https://code-maze.com/dotnet-secure-passwords-bcrypt/>

*Hash PHP.* (s.f.). Obtenido de <https://www.php.net/manual/en/function.password-hash.php>

*IDE Microsoft Visual Studio.* (s.f.). Obtenido de <https://visualstudio.microsoft.com/es/vs/>

*Metodología Cascada.* (s.f.). Obtenido de <https://blog.ganttpro.com/es/metodologia-de-cascada/>

*Metodología Scrum.* (s.f.). Obtenido de <https://proyectosagiles.org/que-es-scrum/>

*Param PHP.* (s.f.). Obtenido de <https://www.php.net/manual/es/mysqli-stmt.bind-param.php>

*ToolTip C#.* (s.f.). Obtenido de <https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.tooltip?view=windowsdesktop-7.0>

*UserControl C#.* (s.f.). Obtenido de <https://learn.microsoft.com/es-es/dotnet/api/system.windows.forms.usercontrol?view=windowsdesktop-8.0>

*Visual Studio Code.* (s.f.). Obtenido de <https://code.visualstudio.com/>

*WinSCP.* (s.f.). Obtenido de <https://winscp.net/eng/index.php>

## 11. Anexos.

- › Pokémon → *Explicación de Pokémon en profundidad.*
- › Páginas que se han mirado para la creación → *Vandal, WikiDex, PokeWiki, etc.*
- › Palabra “Arceus” → Es un Pokémon legendario, el cual se considera Dios de los Pokémon y puede transformar su tipo con unas tablas disponibles para cada tipo existente.
- › Páginas para la creación de los diagramas de la base de datos →
  - » *Diagrama con relaciones y campos.*
  - » *Diagrama Conceptual.*
- › Creación de diagramas → *Visual Paradigm.*
- › Creación DAFO y Organigrama → *Creately.*
- › Creación del instalador de la aplicación → *NSIS* y *NSIS Editor.*
- › Enlace a Github de este mismo proyecto → *Github.*