

Junio de 2024



Pokebarter

Versión 1.0

Sergio González Rojas

ABSTRACTS

Pokebarter es una aplicación web desarrollada con el objetivo de facilitarle a los jugadores de Pokémon un entorno mediante el cual se puedan comunicar para poder realizar el intercambio de pokémon que se encuentren dentro de su colección en los diferentes juegos y que generen un interés mutuo. Los jugadores podrán postear sus pokémon o aquellos en los que estén interesados para darlo a conocer a los demás usuarios. Podrán especificar características como movimientos, juegos de origen, habilidades, sexo, nivel y si es shiny o no. Pokebarter proporciona herramientas para hacer más sencillo el proceso de buscar el pokémon de interés, como filtros por características, perfiles de usuario e imágenes, además proporciona información del propietario de la publicación para poder contactarlo en caso de encontrarse interesado. Es una herramienta ideal para aquellos coleccionistas de Pokémon.

Pokebarter is a web application developed to offer Pokémon players an environment in which they can communicate with each other to trade "Pokémon" into their collections and add them to the different games they are interested in. The users may post their "Pokémon" on their timeline; they could even post Pokémon they want to add to their collection to show other users their interest in those. They can specify Pokémon features, such as moves, origin game, sex, abilities, level, and if it is shiny or not.

Pokebarter provides tools like filters by feature, user profiles and images, to make the search process easier. It also provides post's owner's information to contact them in case of interest. It is the ideal app for Pokémon collectors.

ÍNDICE

ÍNDICE	3
Contenido	3
ÍNDICE DE ILUSTRACIONES	5
3. JUSTIFICACIÓN Y ANTECEDENTES	7
3.1 JUSTIFICACIÓN	7
3.2 ANTECEDENTES	8
4. INTRODUCCIÓN	10
5. OBJETIVOS	11
6. ESTIMACIÓN DE COSTES	12
7. POKEBARTER	14
7.1 TECNOLOGÍA UTILIZADAS	14
7.2 PLANIFICACIÓN	17
7.2.1 Metodología	17
7.2.2 Interfaz	17
7.3 ESPECIFICACIÓN DE REQUISITOS.	20
7.3.1 Requisitos funcionales	21
7.3.2 Requisitos no funcionales	21
7.4 DISEÑO DE LA APLICACIÓN	22
7.5 IMPLEMENTACIÓN Y DOCUMENTACIÓN	24
7.5.1 Arquitectura	24
7.5.2 Estructura	24
7.5.3 Control de versiones	26
7.5.4 Mapa web	27
7.5.5 Estándares	27
7.5.6 Validaciones	28
7.6 PLAN DE PRUEBAS	28
7.6.1 Pruebas de integración	29
7.6.2 Pruebas de sistema	31
7.6.3 Pruebas de aceptación	39
7.6.4 Pruebas de seguridad	42
7.7 DESPLIEGUE Y MANTENIMIENTO.	43
7.7.1 Despliegue	43
7.7.2 Mantenimiento	44
8. CONCLUSIONES Y MEJORAS FUTURAS	46
9. BIOGRAFÍA Y REFERENCIAS	48
10. ANEXOS	49

GLOSARIO DE TÉRMINOS.....	49
IMÁGENES DE LA VERSIÓN MÓVIL.....	51
MANUAL DE USO	¡Error! Marcador no definido.

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Tabla de estimación de costes	11
Ilustración 2 - Sketch de página de inicio	16
Ilustración 3 - Sketch de un post realizado	16
Ilustración 4 - Wireframe de pantalla de inicio de sesión formato PC	17
Ilustración 5 - Wireframe de la pantalla de inicio de sesión formato móvil	17
Ilustración 6 - Wireframe de la pantalla de inicio formato PC	18
Ilustración 7 - Wireframe de la pantalla de inicio formato móvil	18
Ilustración 8 - Diagrama de base de datos	20
Ilustración 9 - Diagrama de flujo	21
Ilustración 10 - Diagrama de actividad para "Crear publicación"	21
Ilustración 11 - Mapa de sitio de Pokebarter	25
Ilustración 12 - Código de modulo consultor de Jest.js	27
Ilustración 13 - Código de testeo con Jest.js	28
Ilustración 14 - Prueba de integración incompatible	28
Ilustración 15 - Datos introducidos correctamente en consulta Jest.js	29
Ilustración 16 - Prueba de integración aprobada	29
Ilustración 17 - Respuesta al autenticarse con una contraseña errónea	30
Ilustración 18 - Respuesta al autenticarse con un username erróneo	30
Ilustración 19 - Ventana de inicio de sesión	30
Ilustración 20 - Pantalla de inicio	31
Ilustración 21 - Formulario de posteo desplegado	31
Ilustración 22 - Aviso de invalidez de pokémon introducido	31
Ilustración 23 - Filtrado de publicaciones de pokémon shiny	32
Ilustración 24 - Filtrado de publicaciones por nombre de pokémon	33
Ilustración 25 - Filtrado de publicaciones de pokémon ofrecidos	33
Ilustración 26 - Filtrado de publicaciones de pokémon solicitado	34
Ilustración 27 - Barra de posteo en perfil propio	34
Ilustración 28 - Perfil ajeno sin barra de posteo	35
Ilustración 29 - Barra de posteo en pantalla de inicio	35
Ilustración 30 - Matriz de pruebas	36
Ilustración 31 - Formulario para iniciar sesión	37
Ilustración 32 - Perfil personalizado del user	37
Ilustración 33 - Botón de cierre de sesión	37
Ilustración 34 - Ventana de edición de información del user	38
Ilustración 35 - Perfil de user ajeno	38
Ilustración 36 - Formulario de posteo	39
Ilustración 37 - Asistencia a los usuarios en formulario de posteo	39
Ilustración 38 - Ventana de confirmación de anulación de post	40
Ilustración 39 - Link de acceso a perfil del propietario de la publicación	40
Ilustración 40 - Publicación de user ajeno	40
Ilustración 41 - Consulta preparada	41
Ilustración 42 - Comando de instalación de Heroku	41
Ilustración 43 - Comando de inicio de sesión de Heroku	41
Ilustración 44 - Contenido de archivo Procfile	42
Ilustración 45 - Comando para iniciar el repositorio local	42
Ilustración 46 - Comando para informar al repositorio que se ha añadido el archivo Procfile	42
Ilustración 47 - Comando para crear el repositorio	42
Ilustración 48 - Comando para desplegar aplicación	42
Ilustración 49 - Pantalla de inicio en el formato móvil	51
Ilustración 50 - Botón de despliegue de laterales	51
Ilustración 51 - Zona de filtrado desplegada en el formato móvil	52

<u>Ilustración 52 - Zona de información personal desplegada en el formato móvil.....</u>	<u>52</u>
--	-----------

3. JUSTIFICACIÓN Y ANTECEDENTES

3.1 JUSTIFICACIÓN

Pokebarter es una iniciativa planteada en base a la numerosa cantidad de miembros que posee la comunidad de Pokémon, más específicamente en este caso a aquellos miembros de la comunidad que en mayor parte se dedican a seguir la franquicia a través de sus juegos. La aplicación en cuestión busca satisfacer las necesidades que en experiencia incluso personal, presentan los jugadores de las múltiples entregas ya sean principales o secundarias que la franquicia de Pokémon ofrece al mundo.

En los juegos de Pokémon, una de las mecánicas más demandadas por los usuarios desde casi el principio de la franquicia es la crianza de las criaturas que presentan, las cuales pueden procrear, evolucionar e intercambiarse por otras, en respecto a esta última característica surge la idea de un espacio para que los jugadores de Pokémon tengan un entorno que les facilite el intercambio de las criaturas que se encuentren en su pertenencia.

Al criar y capturar pokémon en los juegos, es posible conseguirse con algunos cuyas características los posicionen en una escala de mayor valor o cotización, tienden estos a ser más buscados por su nivel de rareza y utilidad, sin embargo se puede dar con casualidad que estos pokémon más raros se crucen en el caminos de algunos jugadores más de una vez con características en común haciéndolos repetidos para el usuario, pero proporcionando a su vez una ventaja, dado que los hace material de intercambio, donde los jugadores podrían obtener a cambio de este otro pokémon que al añadirlo a su colección la haga más valiosa, pero acá es dónde surge la gran interrogante: ¿Con quién puede intercambiar el jugador este pokémon que tiene repetido? La respuesta más acertada a esta pregunta sería: “con otro jugador de Pokémon”, pero ¿y si este usuario no poseyese ningún conocido que jugara los juegos de Pokémon? O, por otro lado, ¿y si los allegados que posee que juegan los juegos de la franquicia no poseen ningún pokémon que sea de interés para el usuario en su colección?

Pokebarter es una aplicación que se crea para prestarle a este tipo de jugadores de Pokémon un servicio que les facilite la tarea de encontrar a alguien que tenga en su poder un pokémon que sea de interés para el usuario y que a su vez el segundo usuario en cuestión esté dispuesto a intercambiar por otro, con una interfaz intuitiva y múltiples caminos para solventar esta problemática presentada para múltiples jugadores de Pokémon.

Originalmente pensado en específico para el juego desarrollado por Pokémon Company junto con Niantic (Pokémon Go), busca en mayor parte presentarse ante los jugadores coleccionistas de pokémon shiny, por lo que esto presentan mayor relevancia en el desarrollo de la aplicación, sin embargo, el alcance de esta se extiende al resto de los juegos de la franquicia, pudiendo presentar pokémon con mayores especificaciones dque las que son requeridas por Pokémon Go.

Es claro que la aplicación va planteada para aquellos jugadores experimentados en los juegos de Pokémon, sin embargo, Pokebarter tratará de ofrecerle a aquello que recién están comenzando su camino en el mundo de los juegos de la franquicia un sendero sencillo de entender, en el cual no haya restricciones con relación a la su alcance, lo cual lo hace particular con respecto a otras aplicaciones cuyo objetivo es similar al de Pokebarter. Además, Pokebarter pretende proporcionar una estructura similar a la de una red social, haciendo al usuario parte de un entorno más gentil y llevadero, en el cual pueda invertir tiempo sin restricciones.

La interfaz de usuario que Pokebarter proporcionaría está en gran parte inspirada en la interfaz de la reconocida e impactante red social Facebook, con la que la mayoría tiende a estar familiarizado, por lo cual presenta un entorno familiar para que todo aquel que la utilice pueda sentirse en capacidad de manejarla.

3.2 ANTECEDENTES

A lo largo de la vida de los juegos de Pokémon se han presentado propuestas para hacer que el disfrute de estos se vea extendido, así como se han ofrecido servicios que han complementado la experiencia del usuario en su trayecto por la aventura que presentan las entregas, ya sean o no directamente relacionadas con los juegos, estas se han convertido en una gran ayuda para los jugadores, y las han permitido reinventar en repetidas ocasiones la forma de desenvolverse en los juegos.

En materia de crianza de pokémon existen numerosos sitios web y foros que instruyen a los entrenadores y criadores pokémon, tal es el caso de webs como WikiDex, que se ha dedicado en conjunto con sus colaboradores a realizar arduas investigaciones de todos los artículos de Pokémon para asistir a sus fanáticos y seguidores, con respecto a los juegos, WikiDex es una fuente bastante útil puesto que presenta información que de gran manera ayuda a aquellos que están centrados en la crianza, en ella se puede encontrar instrucciones para conseguir pokémon de ciertas especies, consejos para mejorarlos, evolucionarlos, es una de las webs más visitadas en relación a Pokémon por la comunidad hispanoablante, teniendo sus versiones en otros idiomas. WikiD3x sin embargo es una web centrada en su totalidad en la información, no otorga directamente a los users la posibilidad de interactuar entre ellos.

Algunos proyectos más orientados a hacer de puente de interacción entre los jugadores de pokemon han sido realizados, como podría ser Poketrade. Esta aplicación web está creada para que los user muestren sus pokémon más valiosos y puedan ser intercambiados, presenta una interfaz sencilla y minimalista que la hace elegante, sin embargo, esta está creada exclusivamente para aquellos jugadores que se centran en

Pokémon Go, imposibilitando participación a los que se dedican a otros juegos a parte de este.

Con respecto a lo planteado con anterioridad, podemos observar que alguno de los temas no abordados puede ser la accesibilidad a todos los juegos de Pokémon hasta la fecha, puesto que el proyecto que más se asemeja al objetivo buscado sólo se extiende hasta Pokémon Go, por lo tanto podría ser de gran ayuda para el resto de jugadores que no se centran sólo en éste, una plataforma que le facilite la comunicación para el intercambio que proponga una interfaz quizás un poco menos minimalista sin perder el recurso de la sencillez.

4. INTRODUCCIÓN

La franquicia de videojuegos de Pokémon comenzó hace 28 años, y desde su nacimiento se ha mantenido en lo alto, acumulando así una enorme cantidad de fans de todas las edades y generaciones, batiendo récords y siendo la estrella en ventas de una de las más grandes industrias de videojuegos como lo es Nintendo. La comunidad de Pokémon se expande cada día, puesto que la compañía se encarga de complacer a sus fanáticos con reiteradas entregas de sus juegos estrella, alcanzando además diferentes plataformas, y usando como estrategia siempre de manera simultánea la innovación y la nostalgia de sus orígenes.

Pese a la gran cantidad de fans, las numerosas comunidades y la enorme cantidad de juegos que la franquicia de Pokémon ha creado con el pasar de los años, la ambición que genera en sus consumidores no ha sido satisfecha, sobre todo en temas referentes a las mecánicas de crianza de las especies, debido a la numerosa cantidad de pokémon que existen en la actualidad, esta se ha complicado, haciéndole imposible conseguirlos a todos por cuenta propia y de manera independiente, esto a obligado a los jugadores de los videojuegos de la franquicia a interactuar con mayor frecuencia entre ellos, sin embargo, a pesar de los múltiples medios que Pokémon facilita a sus usuarios para esto, en algunas ocasiones se puede hacer complicado para algunos de ellos conseguir los pokémon que son tan anhelados para ellos.

Siendo así tan difícil completar la pokedex para aquellos jugadores que desean conseguir a todos los pokémon en todas sus variantes disponibles, se hace necesario optar a caminos externos que impliquen una ayuda para estos. Una alternativa que le permita a los usuarios de los videojuegos de la franquicia comunicarse entre ellos podría mejorar de manera exponencial la experiencia de estos y a su vez facilitar el proceso de colección.

Pokebarter es una aplicación web creada para que los usuarios puedan mostrar de forma directa aquellos Pokémon que están dispuestos a intercambiar con otros jugadores, Cumpliendo el papel de intermediario para la comunicación entre ellos mediante un interés obtenido en lo datos registrados por los usuarios en la aplicación Pokebarter impondrá una experiencia en el juego, en donde el intercambio tomará un nuevo papel protagónico.

5. OBJETIVOS

- Presentar al usuario una aplicación que le permita llevar a cabo de una manera más sencilla el intercambio de pokémon:

Se pretende presentarle al usuario una aplicación web mediante la cual los jugadores de Pokémon puedan observar aquellos pokémon a los que pueda optar a intercambiar según su interés, a su vez se pretende permitirle mostrar aquellos pokémon que están en su poder y aquellos que quiere conseguir mediante los siguientes objetivos específicos:

- Permitirle al usuario presentar los pokémon que posee en su poder para mostrarlos a otros usuarios dentro de la aplicación.
 - Permitirle al usuario presentar los pokémon que son de su interés para notificar dicho interés a los otros jugadores de Pokémon que usan la aplicación.
 - Permitir al usuario tener acceso a los intereses y pokémon poseídos por los otros usuarios que toman parte en la aplicación.
- Proporcionarle al usuario una interfaz de usuario intuitiva, que le permita dar uso a la aplicación de la manera más sencilla posible:

Se quiere presentarle al usuario una interfaz con la que se pueda sentir familiarizado, proporcionándole botones con objetivos claros, formularios fáciles de llenar y una experiencia que le permita interactuar de manera dinámica para no perder la atención del usuario, para ello se plantearán los siguientes objetivos específicos:

- Presentarle al usuario botones con objetivos sencillos y claros de entender para que pueda llevar a cabo su cometido.
- Proporcionarle al usuario formularios que sean fáciles de llenar y cuyo objetivo le sea sencillo de cumplimentar.
- Generar una apariencia estética en la interfaz de usuario para que el usuario se pueda sentir completamente cómodo usando la aplicación.
- Darle al usuario una experiencia interactiva y dinámica para evitar la monotonía y mantener al usuario interesado.

6. ESTIMACIÓN DE COSTES

Para realizar una estimación de costes, en primer lugar, haremos un desglose de tareas para posteriormente estimar su coste económico y en tiempo de producción.

- **Planificación y análisis:** En esta sección se plantean las ideas iniciales del proyecto. Es el tiempo en el que se desarrolla la idea de la aplicación se distribuyen las tareas, se crea y planifica la estructura.
- **Diseño de la interfaz:** Es la etapa en la que entra en juego desarrollo creativo-estético de la interfaz de usuario, en esta se diseñan sketches que den una idea inicial de como lucirá la aplicación frente al usuario, se desarrollan wireframes y maquetas de esta para luego obtener una estimación final de la interfaz final de usuario.
- **Desarrollo:** Es el momento en el que se comienza a codificar. Se desarrolla el back-end de la aplicación, optando al uso de un servidor y de bases de datos. También se contempla el desarrollo del front-end, en el cual aplicamos el diseño planificado, y le damos las funciones que dinamizarán la página. En esta sección también se ve participe la integración con API's y otras tecnologías que sean requeridas.
- **Pruebas:** En este momento se realizan las pruebas de funcionalidad, para ir asegurándose de que el proceso de desarrollo se está ejecutando con éxito en referencia a los objetivos planteados, desarrollándose en paralelo junto a éste.
- **Despliegue y mantenimiento:** En esta etapa es en la que se lanza la aplicación, se pone en uso para el usuario, por lo cual es en donde entran las licencias y dominios. Además, se van corrigiendo errores y a su vez realizando actualizaciones de contenido y de funcionalidades en caso de que estas sean requeridas o deseadas.

Vistas las tareas desglosadas, a continuación, realizaremos una estimación de los recursos necesarios para realizar estas tareas.

- **Costes de personal:** Se estimará el dinero que se invierte en contratar a las personas que se harán cargo de desarrollar las tareas. Dado que sólo se requirió del autor, se expondrá en 0.
- **Herramientas y tecnologías:** Es la etapa en la que se realizara la estimación de costes de los recursos no humanos que se utilizarán para llevar a cabo el desarrollo de la actividad, en este caso se hace referencia a hosting, dominio, y en caso de que incurra en algún tipo de coste IDE's, motores de plantillas u otro tipo de tecnología.

- Costes adicionales: En esta se hace planifican los costes que tendrán los servicios externos no relacionados directamente con la creación de la aplicación, como los costes de marketing y los servicios pagados a terceros.

Tarea	Horas Estimadas	Tarifa por Hora €	Coste €
Planificación	20	50	1.000
Diseño de interfaz	60	50	3.000
Desarrollo front-end	150	65	9.750
Desarrollo back-end	100	70	7.000
Integración de API	24	60	1.440
Pruebas	72	65	4.680
Despliegue	15	45	675
Mantenimiento (1 año)	120	45	5.400
Subtotal Planificación			32.945
Herramienta			Coste €
Hosting (1 año)			180
Dominio (1 año)			15
Subtotal Herramientas			195
Costes adicionales			Coste €
Marketing y SEO			1.400
Asociaciones con terceros			460
Subtotal Costes Adicionales			1.860
Total			35.00

Ilustración 1 - Tabla de estimación de costes

En análisis con lo anteriormente expuesto, podemos concordar en total, de desarrollo de Pokebarter tendría uno 35.000€ de coste en estimación. Cabe destacar que los números son resultados de estimación directa en base a los costes por trabajo obtenidos de la red comparados con las horas que se ha invertido en el desarrollo de la aplicación, que n total fueron 70 días trabajando un estimado de ocho horas por cada uno.

7. POKEBARTER

7.1 TECNOLOGÍA UTILIZADAS

Para crear una aplicación web, es necesario e indispensable el uso de herramientas que faciliten su proceso de codificación, entre estas podemos encontrar por ejemplo los IDE's. A su vez el uso de tecnologías que hagan posible el desarrollo de la misma es imprescindible, puesto que, sin tecnologías como los lenguajes de programación, plugins o lenguajes de marcas, el desarrollo de cualquier aplicación es imposible. Es una parte fundamental dar a conocer todas aquellas que hicieron partícipe en el desarrollo de Pokebarter.

- Visual Studio Code: Es un editor de código fuente desarrollado por Microsoft para Windows, macOS y Linux. Es gratuito y de código abierto, y está diseñado para proporcionar una experiencia de desarrollo ligera y rápida, pero potente y extensible. Aquí hay un desglose de las características y beneficios clave de VS Code. Es conocido por ser rápido y ligero, lo que significa que se carga rápidamente y consume relativamente pocos recursos del sistema. También tiene un amplio ecosistema de extensiones que pueden agregar soporte para nuevos lenguajes de programación, temas, depuración, integración con sistemas de control de versiones, y muchas otras funcionalidades, estas mismas pueden ser utilizadas para crear un entorno de desarrollo personalizado que pueda adaptarse a las necesidades del desarrollador.
- HTML: Es el lenguaje de marcado estándar utilizado para crear páginas web y aplicaciones web. Proporciona la estructura básica de una página web y se utiliza para definir y organizar el contenido que se muestra en un navegador web. HTML utiliza una serie de etiquetas y atributos para estructurar texto, imágenes, enlaces, y otros elementos multimedia en una página web que con una estructurados de manera jerárquica para que los documentos web sean más sencillos de leer.
- CSS: Significa Cascading Style Sheets (Hojas de Estilo en Cascada), es un lenguaje utilizado para describir la presentación de un documento escrito en HTML o XML (incluyendo variantes como SVG, MathML o XHTML). CSS se encarga de cómo se ven los elementos en una página web, controlando aspectos como colores, fuentes, márgenes, bordes, alto y ancho, imágenes de fondo, y la disposición de los elementos.
- Bootstrap: Es un framework de código abierto para el desarrollo de interfaces web, originalmente creado por Twitter. Proporciona una colección de herramientas de diseño, incluyendo estilos CSS y componentes JavaScript, para facilitar la creación de sitios web y aplicaciones web responsivas y móviles.

- **JavaScript:** Es un lenguaje de programación interpretado, de alto nivel y dinámico que se utiliza principalmente para el desarrollo web. Es un lenguaje esencial en el ecosistema de desarrollo web, ya que permite agregar interactividad y funcionalidades dinámicas a las páginas web. Fue desarrollado originalmente por Netscape como un medio para agregar scripts en páginas web, pero desde entonces ha evolucionado para ser utilizado en una variedad de entornos, incluyendo servidores, aplicaciones móviles.
- **EJS:** Su significado es Embedded JavaScript, es un motor de plantillas robusto y flexible para Node.js que facilita la generación de HTML dinámico a partir de datos y lógica JavaScript. Permite a los desarrolladores intercalar código JavaScript directamente dentro de las plantillas HTML, lo que lo convierte en una herramienta poderosa para construir aplicaciones web dinámicas y eficientes.
- **Node.js:** Es un entorno de ejecución de JavaScript del lado del servidor construido sobre el motor V8 de Google Chrome. Permite a los desarrolladores usar JavaScript para escribir aplicaciones del lado del servidor y desarrollar aplicaciones de red escalables y de alto rendimiento. Node.js se ha vuelto extremadamente popular en el desarrollo web debido a su arquitectura basada en eventos y su modelo de E/S (Entrada/Salida) no bloqueante. Usa el famoso sistema MVC que es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.
- **Express.js:** Express.js, a veces también llamado Express, es un framework de back-end Node.js minimalista, rápido y similar a Sinatra, que proporciona características y herramientas robustas para desarrollar aplicaciones de back-end escalables. Ofrece el sistema de enrutamiento y características simplificadas para ampliar el framework con componentes y partes más potentes en función de los casos de uso de tu aplicación. El framework proporciona un conjunto de herramientas para aplicaciones web, peticiones y respuestas HTTP, enrutamiento y middleware para construir y desplegar aplicaciones a gran escala y preparadas para la empresa. También proporciona una herramienta de interfaz de línea de comandos (CLI) llamada Node Package Manager (NPM), donde los desarrolladores pueden obtener paquetes desarrollados. También obliga a los desarrolladores a seguir el principio de No te repitas (DRY).
- **Multer:** Es una librería de middleware para Node.js que se utiliza para manejar datos de formularios en formato multipart/form-data. Este tipo de formato se suele utilizar cuando se envían archivos, como imágenes o documentos, junto con otros datos del

formulario, como cadenas de texto. Algunos ejemplos incluyen la carga de avatares de usuarios, imágenes de productos o archivos adjuntos en correos electrónicos.

- **AJAX:** Significa Asynchronous Javascript y XML. Aunque no es un lenguaje de programación, AJAX permite a los desarrolladores construir aplicaciones web más eficientes, rápidas e interactivas con herramientas como HTML, CSS, JavaScript y XML. Hace referencia a una serie de tecnologías útiles para desarrollar aplicaciones web. Es un método que permite al usuario de una aplicación web interactuar con un sitio sin necesidad de actualizar constantemente las páginas cada vez que realiza un cambio de entrada.
- **MySQL:** Es un sistema de gestión de bases de datos muy popularizado especialmente en entornos web, debido a su confiabilidad, facilidad de uso y rendimiento. Es creado y mantenido por Oracle Corporation y posee con código abierto.
- **Morgan:** Es un middleware para la creación de registros de solicitudes HTTP en aplicaciones web Node.js. Es una biblioteca de registro de solicitudes HTTP que se utiliza comúnmente en aplicaciones Express.js, aunque también se puede utilizar con otros frameworks web de Node.js.
- **jQuery:** Es una biblioteca de JavaScript rápida, pequeña y de uso general que simplifica la manipulación del DOM y la interacción con el navegador web.
- **Nodemon:** Es una herramienta de desarrollo para aplicaciones Node.js que supervisa los archivos en un proyecto y automáticamente reinicia la aplicación cuando detecta cambios en el código fuente. Esto elimina la necesidad de detener y reiniciar manualmente la aplicación cada vez que se realizan cambios en los archivos.
- **SweetAlert2:** Es una biblioteca de JavaScript para crear fácilmente diálogos de alerta, confirmación y diálogos modales personalizados en páginas web.
- **Jest.js:** Es un framework de pruebas para JavaScript, especialmente utilizado en entornos de desarrollo de aplicaciones web y Node.js.
- **mysql2:** es una biblioteca para Node.js que permite interactuar con bases de datos MySQL y MariaDB. Es una versión mejorada y moderna de la biblioteca mysql original, con mejoras en el rendimiento y la compatibilidad con las promesas de JavaScript.
- **Git:** Es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en el mismo proyecto simultáneamente sin conflictos y mantiene un historial de todos los cambios realizados en el código fuente.
- **GitHub:** Es una plataforma de alojamiento de código que utiliza el sistema de control de versiones Git. Proporciona una serie de características y herramientas que facilitan la colaboración entre desarrolladores y la gestión de proyectos de software.

- Heroku: es una plataforma como servicio (PaaS) que permite a los desarrolladores construir, ejecutar y operar aplicaciones completamente en la nube. Ofrece un entorno gestionado donde los desarrolladores pueden desplegar aplicaciones sin preocuparse por la infraestructura subyacente.

7.2 PLANIFICACIÓN

7.2.1 Metodología

Al momento de desarrollar una aplicación, la planificación de esta es importante, puesto que puede ayudar a gran medida a no perder el rumbo al objetivo que se quiere alcanzar, para esto es recomendable utilizar las metodologías de desarrollo de software, es muy importante escoger la que más se adapte al estilo de trabajo y al proyecto que pretende realizarse ya que mediante esta el trabajo de desarrollo se realizará con mayor eficiencia.

En el proceso de desarrollo de Pokebarter se pretende utilizar la metodología de desarrollo en espiral que es una combinación entre el modelo en cascada y un modelo por iteraciones.

El proceso pasa por distintas etapas, desde la de conceptualización, siguiendo el desarrollo, luego una fase de mejoras, para finalizar con el mantenimiento. Dentro de cada etapa, se tiene una serie de fases que transcurren desde la planificación, pasando por el análisis de riesgos, el desarrollo y finalizando en la evaluación de lo realizado. Se incorpora también una fase de enlace entre etapas, para facilitar la transición entre las mismas.

En este proceso se comienza planteando una serie de pequeños requisitos, al finalizar estos se plantea un nuevo conjunto de requisitos, con los cuales se pasa de nuevo por todas las etapas hasta completarlas y así sucesivamente. Se agrega la funcionalidad para el requerimiento adicional en espirales cada vez mayores, hasta que la aplicación está lista para la fase de producción.

7.2.2 Interfaz

Al realizar una aplicación web, sin duda el diseño de la misma tiene casi el mismo nivel de importancia que la propia funcionalidad, por esa razón es importante mantener siempre los estándares de diseño cubiertos, para ello existen previos pasos a la realización del mismo donde se utilizan metodologías ágiles para plantearlo con decoro. En Pokebarter antes de realizar el diseño se crearon sketches y wireframes que aportaran una idea más clara de lo que la aplicación pretendía mostrar y cómo lo haría.

- Sketch: Es un dibujo rápido y sencillo que captura la idea básica de un diseño. En esta etapa, se planteaba como sería la interfaz principal de la pantalla de inicio de Pokebarter.

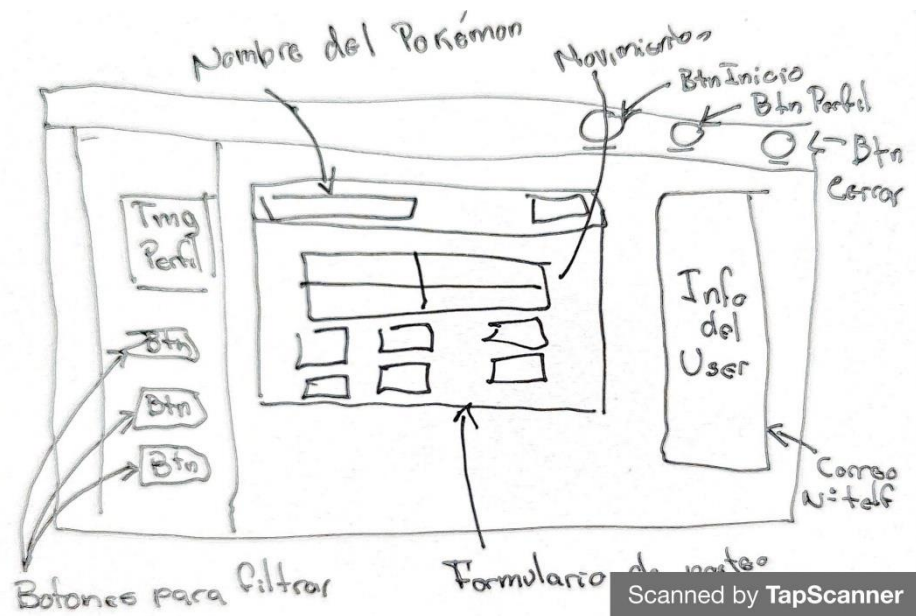


Ilustración 2 - Sketch de página de inicio.

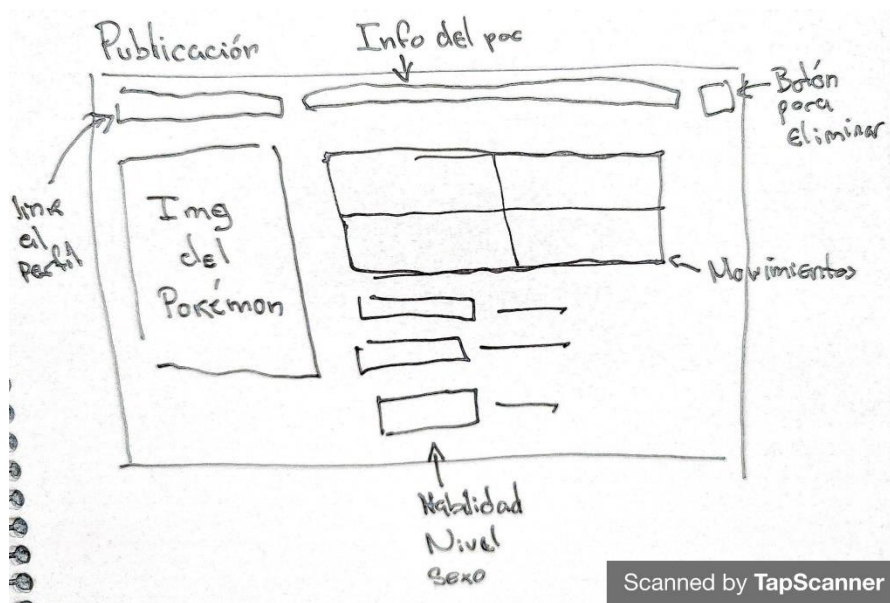


Ilustración 3 - Sketch de un post realizado.

- Wireframe: Es una representación esquemática de la estructura básica de una página web o aplicación, enfocada en el layout y la disposición de elementos. En esta etapa del desarrollo del diseño de Pokebarter se plantea el diseño del formato móvil.

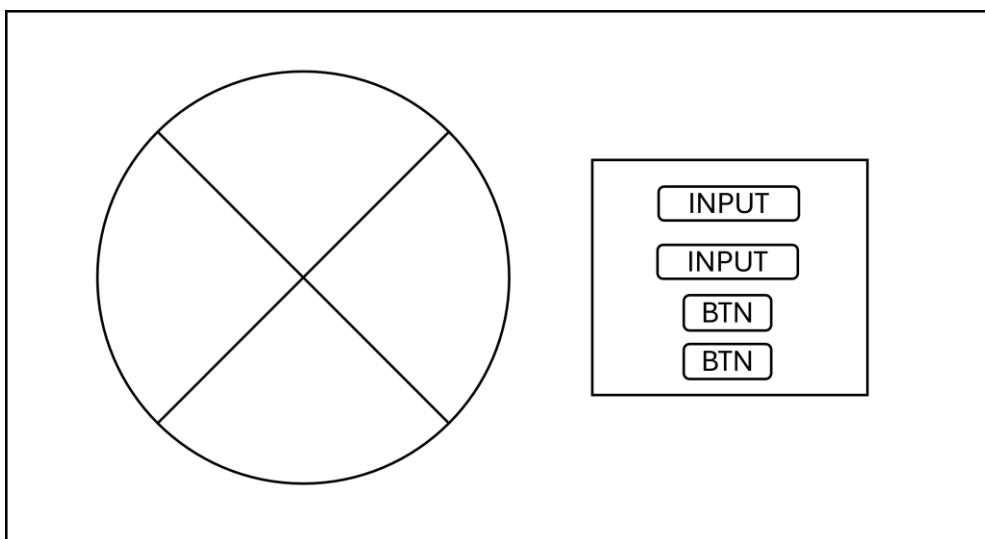


Ilustración 4 - Wireframe de pantalla de inicio de sesión formato PC.

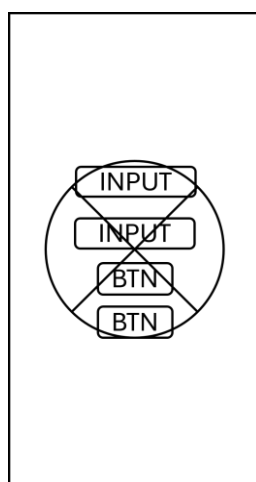


Ilustración 5 - Wireframe de la pantalla de inicio de sesión formato móvil.

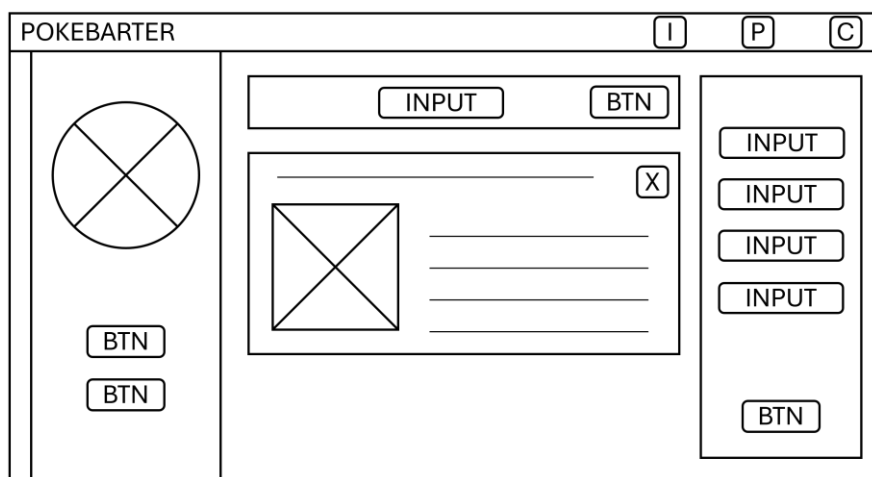


Ilustración 6 - Wireframe de la pantalla de inicio formato PC

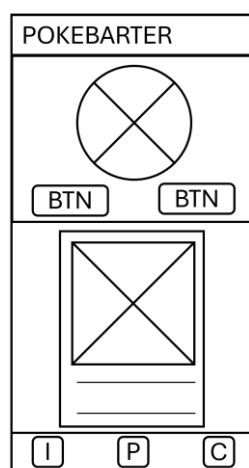


Ilustración 7 - Wireframe de la pantalla de inicio formato móvil.

7.3 ESPECIFICACIÓN DE REQUISITOS.

Durante el desarrollo de una aplicación siempre es importante plantearse los que serán los requisitos que está debe cumplir, estos deben estar firmemente basados en los objetivos que requiere cumplir la misma, es decir deben ponerse en claro las metas que se deben pautar para que la aplicación cumpla con los objetivos funcionales que esta pretende alcanzar, sin embargo el cómo la aplicación funcione no es el único punto que debe tomarse en cuenta a la hora de plantearse los requisitos, los llamados requisitos no funcionales también son indispensables para que el proyecto desarrollado tenga éxito, así que es necesario además tener claro las funciones que la aplicación debe tener, es indispensable tomar en cuenta aspectos como vendrían siendo a dónde la aplicación quiere llegar o qué quiere proyectar.

7.3.1 Requisitos funcionales

- Manejo de sesiones:
 - El usuario debe ser capaz de iniciar sesión y tener acceso a toda la información que la aplicación debe brindarle.
 - El usuario debe ser capaz de cerrar sesión sin perder la información que esta sesión generó.
 - Los usuarios deben poder crear perfiles personales, proporcionando información básica como nombre, foto de perfil, contacto e información que pueda ser considerada pertinente.
- Creación y gestión de perfiles:
 - La aplicación debe permitir a los usuarios editar y actualizar sus perfiles en cualquier momento, incluyendo la modificación de la información personal.
 - El usuario debe ser capaz de acceder a todos los perfiles de los usuarios que pueda ver a través de la aplicación
- Publicación de contenido:
 - Los usuarios deben poder crear publicaciones en las cuales se pueda ver la información de pokémon que desea mostrarse especificando la información pertinente del mismo, para que así pueda ser percibida adecuadamente.
 - La aplicación debe proporcionarle al usuario una ayuda a la hora de realizar una publicación para que pueda publicar el contenido correctamente.
 - Los usuarios deben tener permitido eliminar sus propias publicaciones y sólo las suyas propias.
- Gestión de contenido de la aplicación:
 - Los usuarios deben ser capaces de ver las publicaciones de los demás usuarios y corroborar qué usuario realizó dicha publicación
 - La aplicación debe proporcionar al usuario una manera práctica de acceder a las publicaciones para que este pueda dirigirse directamente a aquellas que le generen interés.

7.3.2 Requisitos no funcionales

- La aplicación debe proporcionar una interfaz clara, amigable e intuitiva para el usuario.
- La aplicación debe ser escalable y capaz de manejar a varios usuarios simultáneamente.

- La aplicación debe ser rápida en respuestas, y usar una estructura que minimice el tiempo de inactividad

7.4 DISEÑO DE LA APLICACIÓN

En el diseño de toda aplicación es importante plantearse el contenido que está tendrá y las funciones que implementará, sin embargo, para esto también es necesario planear el recorrido de los procesos que la aplicación llevará a cabo.

Una aplicación que registre datos dependerá siempre de una base de datos que los pueda almacenar, por lo tanto, siempre es importante plantearse cómo esta estará estructurada. Pokebarter utiliza una base de datos que consta de dos tablas, unidas entre sí mediante una relación de uno a muchos, una de estas es la tabla que se encarga de registrar a los usuarios de la aplicación, mientras que la otra se encarga de registrar las publicaciones que cada uno de estos realiza, esta última consta de una clave foránea perteneciente a la primera que es el identificador del usuario, por lo cual se da a entender que un usuario puede tener muchas publicaciones, más una publicación pertenece exclusivamente a un solo usuario

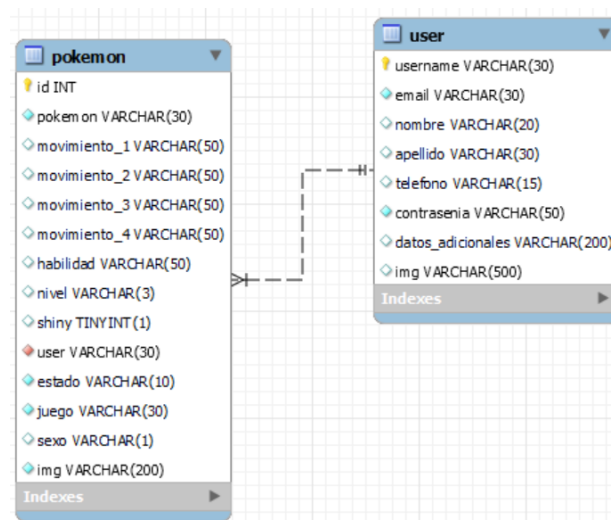


Ilustración 8 - Diagrama de base de datos.

En un diagrama de flujo se muestra el recorrido que realiza la aplicación en consecuencia a las acciones que realiza el usuario. En este caso sirve para saber cómo el usuario se puede mover entre las páginas y las acciones que este puede realizar en cada una de ellas. Se puede observar que existen cuatro páginas principales a las cuales se puede acceder entre sí, a su vez, desde éstas pueden ejecutarse funcionalidades diferentes, cada una depende de la página en la que se encuentre.

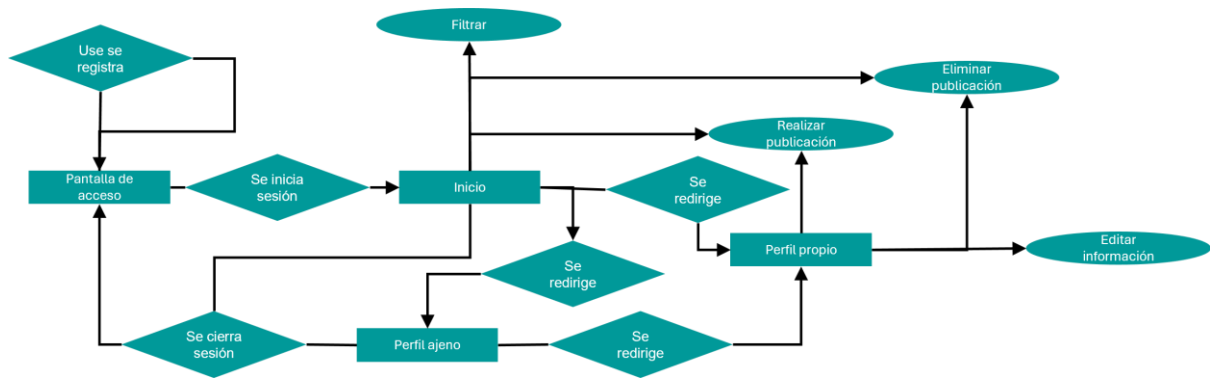


Ilustración 9 - Diagrama de flujo

El diagrama de actividad es utilizado para indicar el camino que sigue un proceso mediante las diferentes etapas por las cuales este pasa. Permite plantear el procedimiento adecuado de las más importantes funcionalidades de la aplicación.

En Pokebarter podemos estudiar el proceso en el cual un usuario realiza una publicación, se puede observar que una actividad desencadena una posterior, y cada una de ellas es ejecutada en una diferente etapa, en este caso se contemplaron como etapas la interacción del user, la vista, el controlador y el modelo, cada uno encargado de tareas diferentes a realizar.

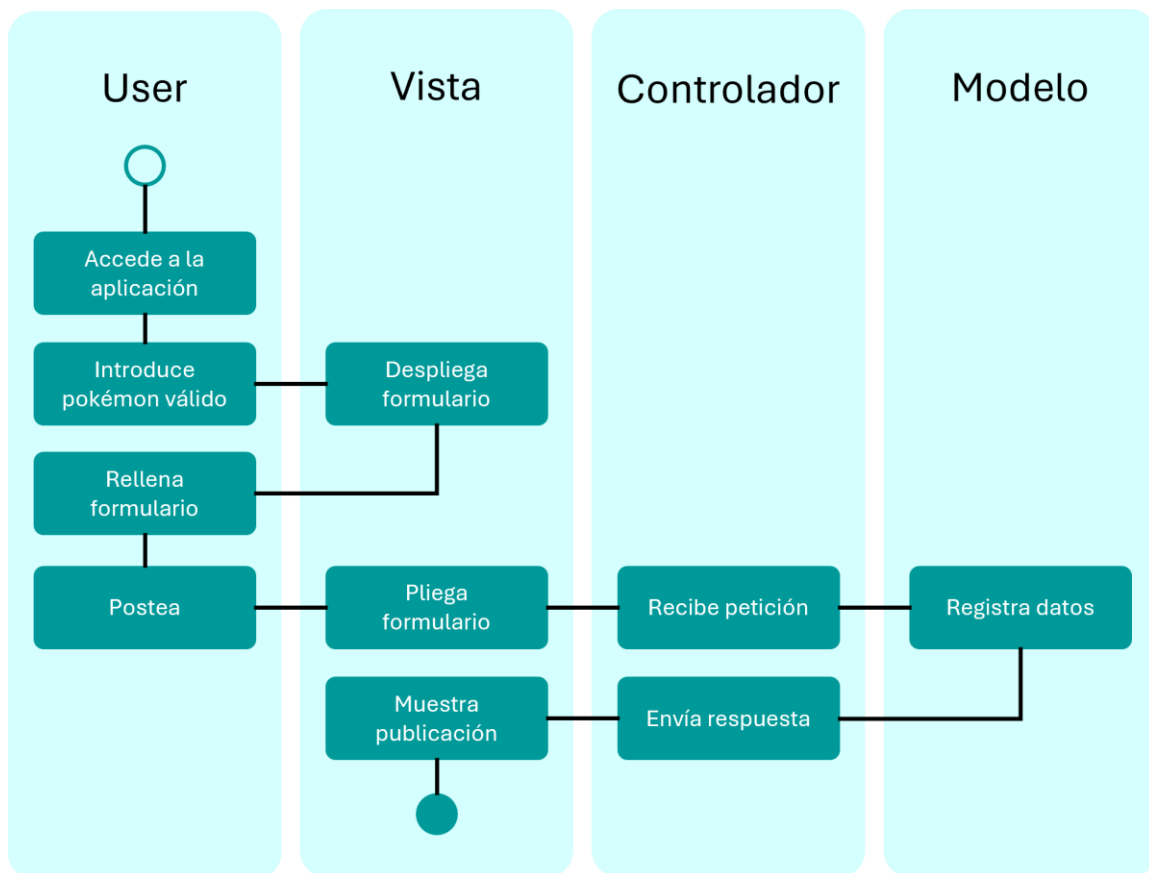


Ilustración 10 - Diagrama de actividad para "Crear publicación".

7.5 IMPLEMENTACIÓN Y DOCUMENTACIÓN

7.5.1 Arquitectura

Cada aplicación desarrollada posee un diseño particular llamado arquitectura de software, este se caracteriza por definir cómo los componentes de la misma interactúan entre sí, cómo se organizan y cómo son capaces de cumplir con los requerimientos funcionales y no funcionales, esta también proporciona una visión global del sistema, lo que permite a los desarrolladores y arquitectos comprender su estructura y tomar decisiones informadas durante el proceso de desarrollo.

La arquitectura de software es fundamental para el desarrollo de una aplicación ya que con esta se puede obtener mayor rendimiento y escalabilidad, garantizando de esta manera el desarrollo de una aplicación de calidad.

Modelo-Vista-Controlador: Esta arquitectura también conocida como MVC es caracteriza por estar dividida en tres capas, la primera y más cercana al usuario es la vista, que es aquella capa que actúa como interfaz de usuario y se encarga de mostrar la información, en la capa del medio se encuentra la capa del controlador, esta es la encargada de conectar la vista con el modelo, es en donde se realizan las peticiones mediante comandos

Ilustración 5 - Diagrama de actividades.

a las que la última capa, el modelo, dará respuesta, sirviendo así la información solicitada por el controlador y enviándola para que pueda ser plasmada en la vista.

Siendo Pokebarter una aplicación desarrollada utilizando Node.js y Express.js esta cuenta con esta arquitectura, en la cual el controlador está dividido en dos partes, una que procesa las peticiones realizadas por el cliente, y la otra que es la que se encarga de codificar estas peticiones y realizar la consulta pertinente al servidor o modelo, en este caso particular.

7.5.2 Estructura

Durante el desarrollo del código se genera una estructura jerárquica de archivos y directorios, que mediante su disposición se encargan de mantener el código ordenado y disponible con facilidad en caso de acceso. Para la creación de Pokebarter se crea una estructura de carpetas anidadas en una carpeta llamada Pokebarter cuya única función es unificar las mismas.

- Pokebarter/
 - .svscode/: Carpeta que guarda la configuración del proyecto.
 - database/: Carpeta que contiene el script de la base de datos de Pokebarter.
 - node_modules/: Almacena todas las dependencias y módulos necesarios para que la aplicación funcione correctamente. Aquí se incluyen tanto las dependencias

directas que son aquellas que son especificadas en el archivo package.json del proyecto, y también las dependencias transitivas.

- src/: Contiene el código fuente de la aplicación.
 - controllers/: En esta carpeta se encuentra el controlador, la capa encargada de mantener comunicadas la vista y el modelo.
 - CustomerController.js
 - public/: En esta carpeta se encuentran almacenadas las carpetas en donde se almacenan las imágenes, los archivos CSS y los scripts utilizados para dinamizar la aplicación en el lado cliente.
 - css/: Contiene las hojas de estilo, la asociada a Bootstrap, y la personalizada.
 - styles.css
 - stilesSolar.css
 - img/: Esta contiene las imágenes que usará la aplicación. Contiene una carpeta en la que se registran todas las imágenes que los users eligen de perfil a la hora de registrarse y otros archivos con las imágenes de utilidad.
 - ImagenPerfil/: Carpeta que guarda las imágenes de perfil de los users.
 - Archivos de imágenes.
 - scripts/: Contiene todos los archivos .js encargados de animar y dinamizar la aplicación, contiene uno por cada vista que lo requiera.
 - Pruebas: Carpeta con los scripts que se usaron para hacer las pruebas de integración:
 - prueba.test.js: Módulos que se usan al realizar la prueba.
 - pruebaIntegracion.js: Script que ejecuta las pruebas.
 - inicio.js
 - lateral.js
 - login.js
 - muro.js
 - nav.js
 - perfil.js
 - posteo.js

- routes/: Carpeta que contiene el enrutador.
 - customer.js: Encargado de procesar las peticiones realizadas desde el cliente y redirigirlas al controlador de manera adecuada.
- views/: Contiene las vistas.
 - Partials/: Carpeta que guarda las vistas que no constituyen una página en sí mismas, sino que son fragmentos de las páginas principales.
 - _footer.ejs
 - _header.ejs
 - _lateral.ejs
 - _listas.ejs
 - _muro.ejs
 - _nav.ejs
 - _posteo.ejs
 - _profile.ejs
 - inicio.ejs
 - login.ejs
 - perfil.ejs
- app: Es el modelo de la aplicación, constituye la parte del servidor de Pokebarter.
 - package-lock.json: Archivo que almacena la versión exacta de todas las dependencias del proyecto.
 - package.json: Archivo que contiene metadatos sobre el proyecto, como su nombre, versión, descripción, scripts de ejecución, dependencias del proyecto, entre otros.

7.5.3 Control de versiones

Este es el sistema que registra todos los cambios realizados en los archivos que contienen el código de la aplicación, al realizar el control de versiones, se posee la capacidad de rastrear y revertir los cambios que se hayan realizado en este debido a que registra un historial de los mismos.

Para realizar el control de versiones en Pokebarter se usa la herramienta Git. A través del sistema de repositorios que esta brinda, se puede mantener bajo control el sistema de actualizaciones. Los múltiples comandos que posee Git facilitan la gestión.

- Git clone: Permite el descargar una copia del repositorio de Pokebarter subido a Git y poder realizar cambios en el con la intención de realizar mejoras sin arriesgar la versión ya adaptada de la aplicación.

- Git add: Es el comando con el cual se le notifica al repositorio de Git que se ha añadido algún nuevo archivo. Fue utilizado en múltiples ocasiones cuando se creaban los nuevos scripts que dieran funcionalidad a los diferentes archivos generados por el motor de plantillas ejs.

- Git commit: Este realiza un guardado de los cambios realizados en el repositorio. Al editar el código fuente del repositorio clonado, se hacía commit para registrar los cambios realizados.

7.5.4 Mapa web

Los mapas web son una parte importante del diseño de una aplicación debido a que con estos nos podemos orientar con relación a lo que las diferentes páginas de la misma mostrarán u ofrecerán. Pokebarter es una aplicación que no cuentas con una diferencia altamente notable en sus páginas, debido a que en principio sólo tiene tres, la página de login,

11

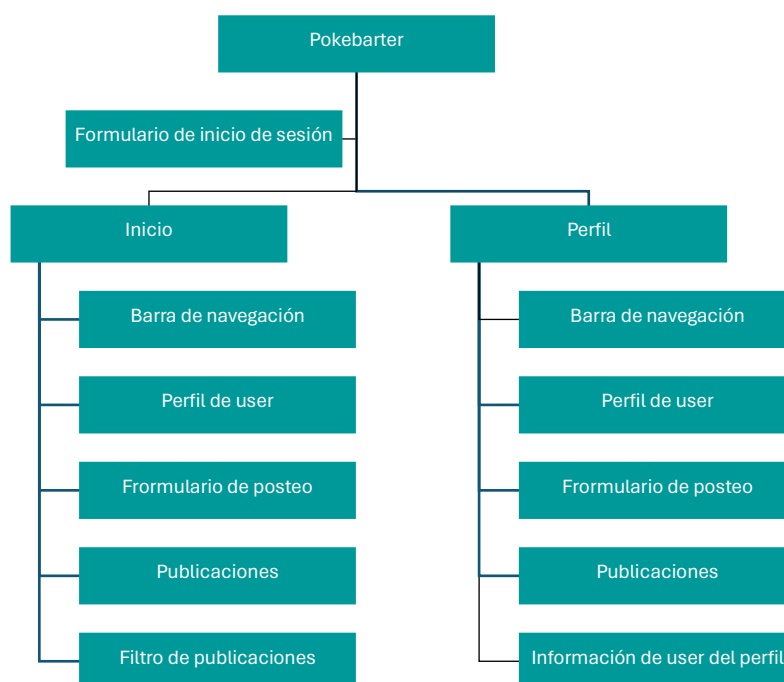


Ilustración 11 - Mapa de sitio de Pokebarter.

7.5.5 Estándares

En el proceso de creación de una aplicación, se utilizan ciertos protocolos que describen el comportamiento que tendrá la World Wide Web, estos son conocidos como estándares. Tienden a ser caracterizados por ser abiertos, y les permiten a todos a los sitios web interoperar con las distintas herramientas con las que interactúan. Se presentan en forma

de especificaciones técnicas dirigidas. El objetivo de estos es permitirles a los navegadores ejecutar un sitio web sin necesidad de una estructura exclusiva para este. Los estándares implementados en Pokebarter permiten desarrollar su estructura de forma práctica y eficiente.

- **HTML:** Es utilizado para darle al documento una estructura. Como se explicó con anterioridad, ese proporciona una estructura jerárquica, de esta manera, los elementos de la aplicación pueden tener una estructura definida. El esqueleto de Pokebarter está definido en documentos EJS, pero en estos el lenguaje de HTML se ve embebido.
- **CSS:** Con este la aplicación cobra un estilo vivido, dando a la interfaz un sentido estético para hacerla más atractiva ante el usuario.
- **JavaScript:** Es el lenguaje de programación utilizado para dinamizar la aplicación y construir tanto el lado cliente como el lado servidor.
- **JSON:** Este es el formato en el que son recibidos y enviados los cuerpos de las peticiones realizadas dentro de Pokebarter, como aquellas que se hacen al servidor o a la PokeAPI.
- **MIME:** Este estándar es utilizado para procesar los archivos que son enviados por el usuario a la hora de seleccionar su foto de perfil en el momento de registrarse.

7.5.6 Validaciones

Dentro de una aplicación se realizan mediante la implementación de reglas y criterios que aseguran que los datos ingresados por los usuarios sean correctos, coherentes y seguros. Estas validaciones se llevan a cabo en las diferentes capas de la aplicación, el front-end y el back-end, así como también se pueden realizar validaciones que observen la interacción entre estos. Para garantizar el correcto funcionamiento de Pokebarter se han realizado validaciones en todos estos aspectos con un resultado que ha debido perfeccionarse en reiteradas ocasiones hasta obtener el resultado esperado en las pruebas realizadas en el plan de pruebas en el plan de pruebas se

7.6 PLAN DE PRUEBAS

Para comprobar que las aplicaciones funciones adecuadamente y cumplan con los requerimientos, es necesario realizar pruebas que demuestren esto, o en caso contrario, que notifiquen que hay un error para que así podamos solucionarlo. Para esto se realizan diferentes tipos de pruebas, que en conjunto garantizan que la aplicación cumple con sus objetivos, debido a que Pokebarter no se maneja con clases, las pruebas unitarias no fueron realizadas, por lo tanto, se pasó directamente a ejecutar las pruebas de integración, a su vez, debido al entorno en el que se ejecuta la aplicación, tampoco se darán a cabo pruebas de rendimiento.

7.6.1 Pruebas de integración

Estas son pruebas que se realizan para asegurarse de que varias partes de la aplicación funcionan de manera adecuada en conjunto. Para realizar estas pruebas en Pokebarter se decidió demostrar que las parte cliente está adecuadamente integrada con la parte del servidor, demostrado que las consultas realizadas obtienen los datos esperados.

Para realizar las pruebas de integración se utilizó mysql2 junto con Jest.js. Se desarrolló un módulo sencillo que hiciera conexión con la base de datos

```
const mysql = require('mysql2/promise');

let connection;

async function connect() {
  connection = await mysql.createConnection({
    host: "localhost",
    user: "root",
    password: "1234",
    database: "pokebarter"
  });
}

async function close() {
  if (connection) {
    await connection.end();
  }
}

async function getUserByUsername(username) {
  const [user] = await connection.execute(
    'SELECT * FROM user WHERE username = ?', [username]
  );
  return user[0];
}

module.exports = { connect, close, getUserByUsername };
```

Ilustración 12 - Código de modulo consultor de Jest.js.

Posteriormente se creó otro módulo que realizara las consultas pertinentes, utilizando Jest.js para comparar los resultados obtenidos con los resultados esperados.

```

const { connect, close, getUserByUsername } = require('../purebasPokebarter');

beforeAll(async () => {
  await connect();
});

afterAll(async () => {
  await close();
});

test('Obtener un usuario', async () => {
  const userId = "seogonzalez"
  const user = await getUserByUsername(userId);

  expect(user).toEqual({ username: userId, email: 'sergioagrvi@gmail.com' });
});

```

Ilustración 13 - Código de testeo con Jest.js

Al ejecutar el comando `npm run test` (debido a que en el `package.json` definimos que con “test” se accedería al framework de pruebas) podemos observar los resultados. En caso de que los resultados no coincidan con lo esperado indicará el error.

```

expect(received).toEqual(expected) // deep equality

- Expected   - 0
+ Received   + 6

Object {
+   "apellido": "González Rojas",
+   "contrasenia": "1234",
+   "datos_adicionales": "",
  "email": "sergioagrvi@gmail.com",
+   "img": "1716389693391-1711147417044.jpg",
+   "nombre": "Sergio",
+   "telefono": "654587900",
  "username": "seogonzalez",
}

```

Ilustración 14 - Prueba de integración incompatible

En caso contrario, cuando es modificado el resultado esperado para que sea igual al obtenido el resultado es validado:

```
test('Obtener un usuario', async () => {
  const userId = "seogonzalez"
  const user = await getUserByUsername(userId);

  expect(user).toEqual({
    "apellido": "González Rojas",
    "contrasenia": "1234",
    "datos_adicionales": "",
    "email": "sergioagrvi@gmail.com",
    "img": "1716389693391-1711147417044.jpg",
    "nombre": "Sergio",
    "telefono": "654587900",
    "username": "seogonzalez"});
});
```

Ilustración 15 - Datos introducidos correctamente en consulta Jest.js

```
PS C:\Users\sergi\Desktop\Pokebarter> npm run test

> tfg@1.0.0 test
> jest

PASS src/public/scripts/pruebas/prueba.test.js
  ✓ Obtener un usuario (10 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 0.526 s, estimated 1 s
Ran all test suites.
PS C:\Users\sergi\Desktop\Pokebarter>
```

Ilustración 16 - Prueba de integración aprobada.

7.6.2 Pruebas de sistema

Estas son las pruebas que evalúan el comportamiento de todo el sistema desde la perspectiva de los usuarios, se realizan desde la aplicación simulando el comportamiento del mismo en casos de uso que deben definirse con antelación.

Para Pokebarter se definieron cuatro escenarios de prueba:

- Inicio de sesión: En este escenario se plantearán tres casos de prueba tomando en cuenta que el user es seogonzalez y la contraseña será 1234.
 - Introduciendo un user correcto y una clave incorrecta.

seogonzalez

.....

Iniciar sesión

Los datos introducidos no son correctos

Registrarse

Ilustración 17 - Respuesta al autenticarse con una contraseña errónea.

- Introduciendo un user incorrecto y una clave correcta.

azsxdcrtvgbhnjm

....

Iniciar sesión

Los datos introducidos no son correctos

Registrarse

Ilustración 18 - Respuesta al autenticarse con un username erróneo.

- Introduciendo un user correcto y una clave correcta.

seogonzalez

....

Iniciar sesión

Registrarse

Ilustración 19 - Ventana de inicio de sesión

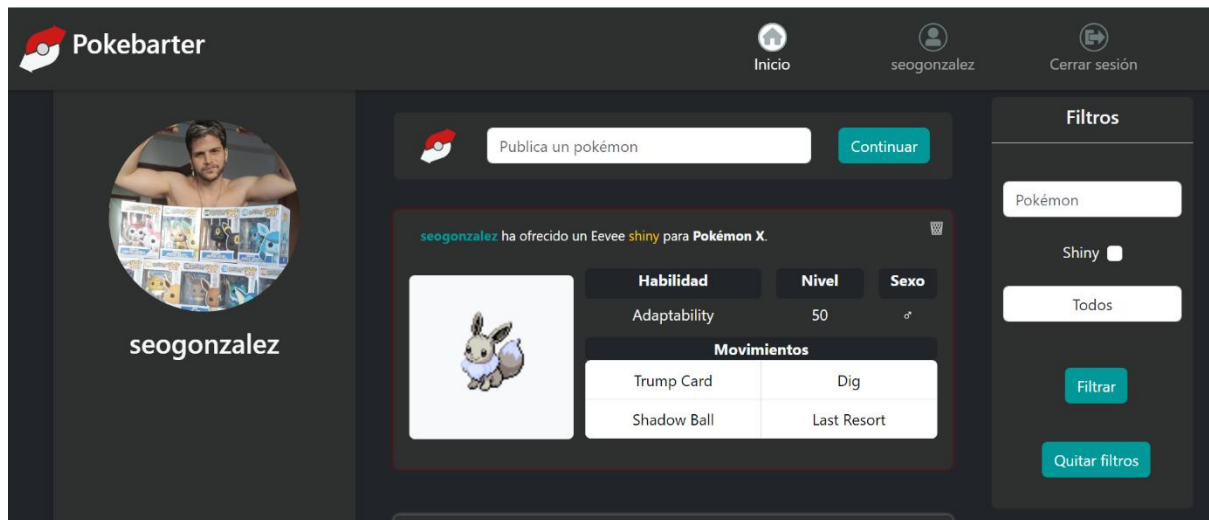


Ilustración 20 - Pantalla de inicio.

- Publicar un pokémon: En este caso sólo se tendrán en cuenta dos casos
 - Se introduce el nombre de un pokémon correcto y se presiona continuar:

Ilustración 21 - Formulario de posteo desplegado.

- Se introduce el nombre de un pokémon incorrecto y se presiona continuar:

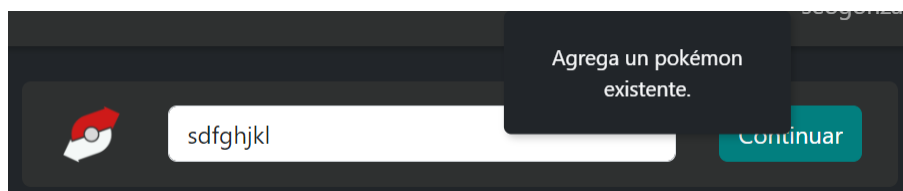


Ilustración 22 - Aviso de invalidez de pokémon introducido.

- Filtrado de pokémon: Debido a que las variantes de este escenario pueden ser demasiadas, sólo se tendrán en cuenta cuatro casos de prueba:
 - Se filtra por shiny:

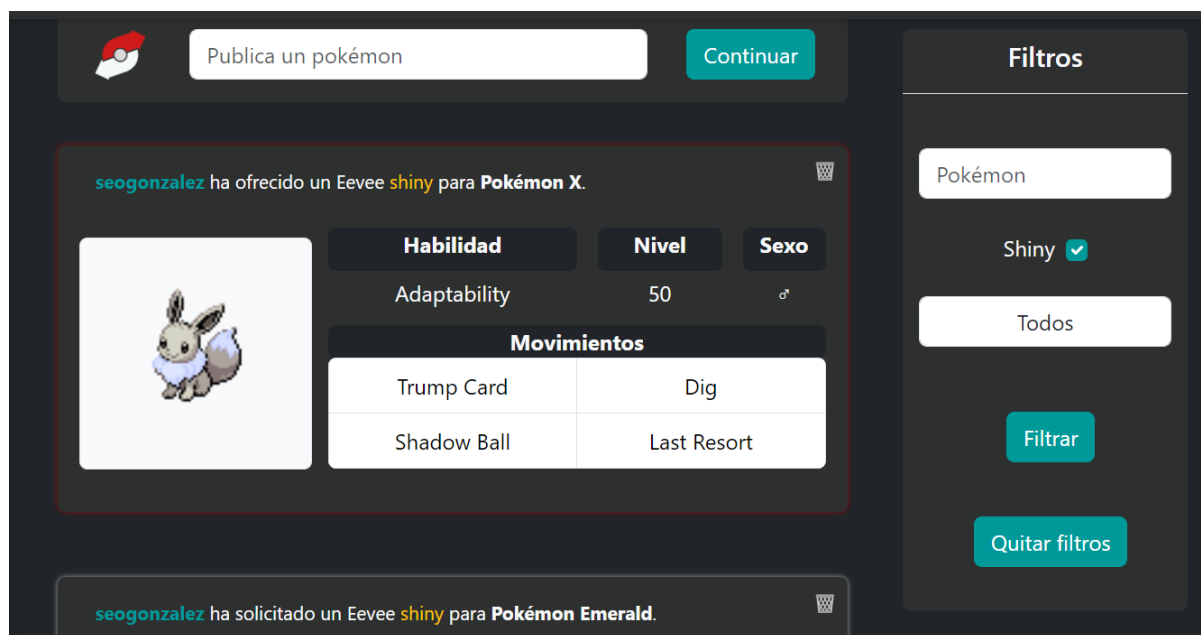


Ilustración 23 - Filtrado de publicaciones de pokémon shiny.

- Se filtra por nombre pokémon:

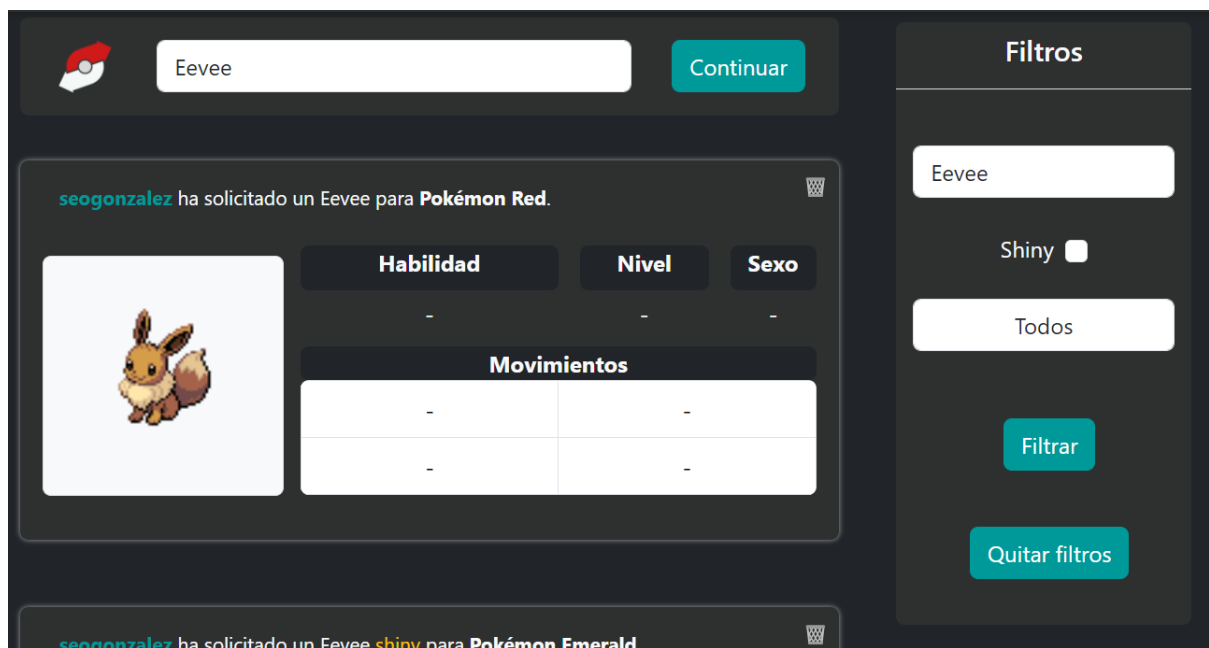


Ilustración 24 - Filtrado de publicaciones por nombre de pokémon.

- Se filtra con estado ofrecido:

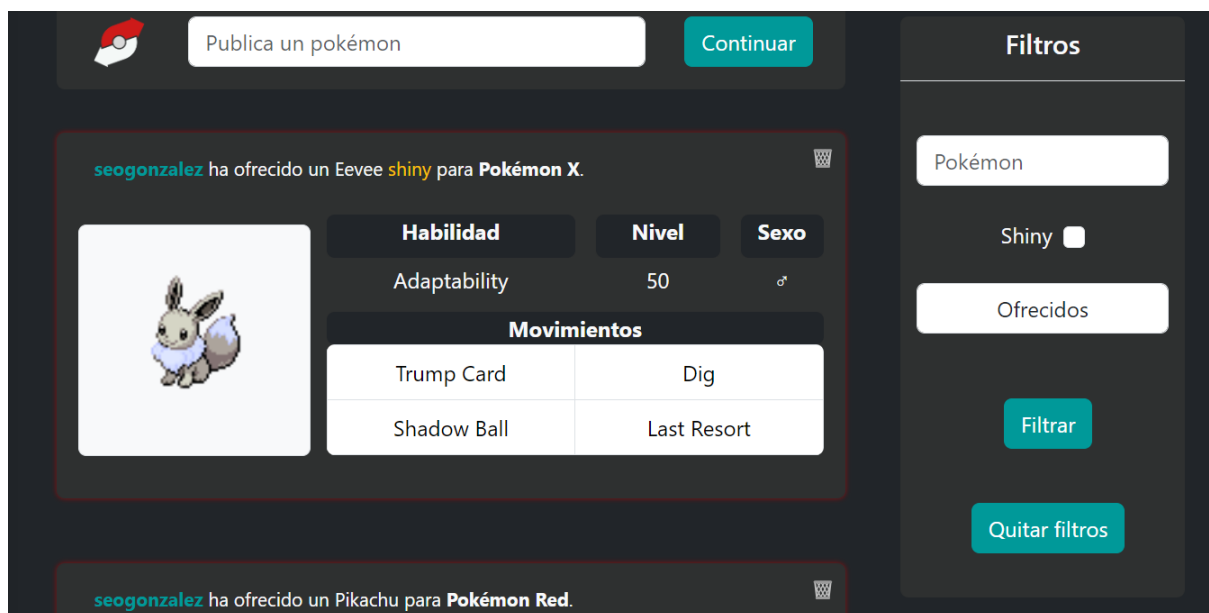


Ilustración 25 - Filtrado de publicaciones de pokémon ofrecidos.

- Se filtra con estado solicitado:

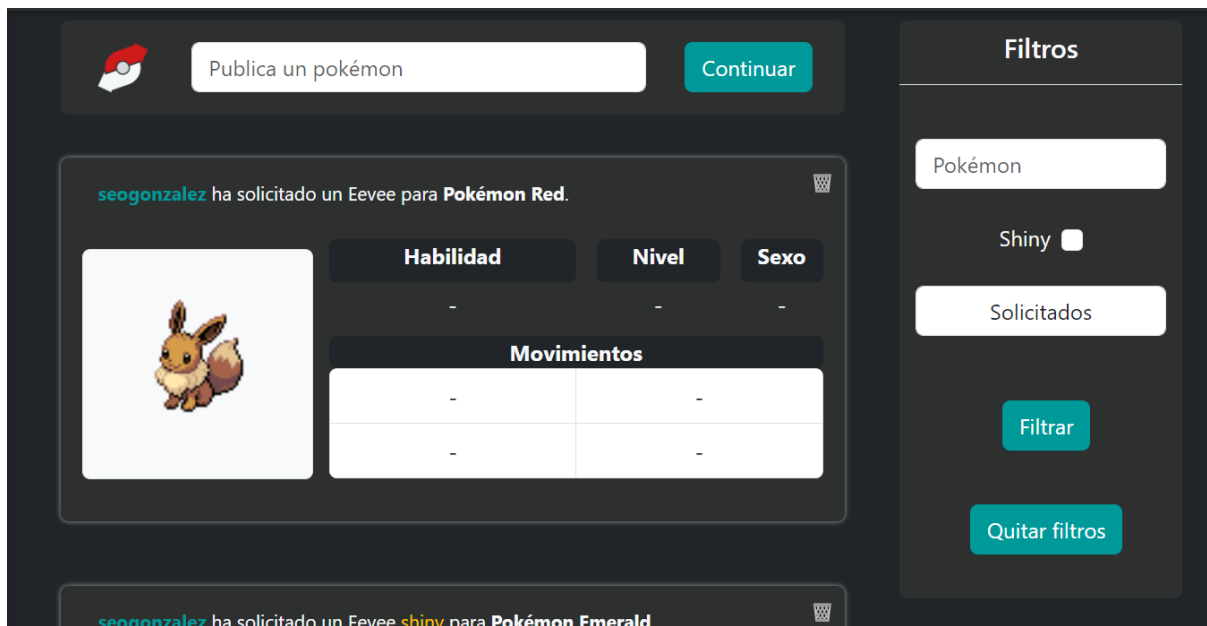


Ilustración 26 - Filtrado de publicaciones de pokémon solicitado.

- Manejo de publicación: En este escenario de prueba se comprobará si se accede a los perfiles con la intención de demostrar que sólo se puede realizar una publicación desde el perfil propio o desde el inicio, para esto se plantearán tres casos.
 - Se accede al perfil propio:

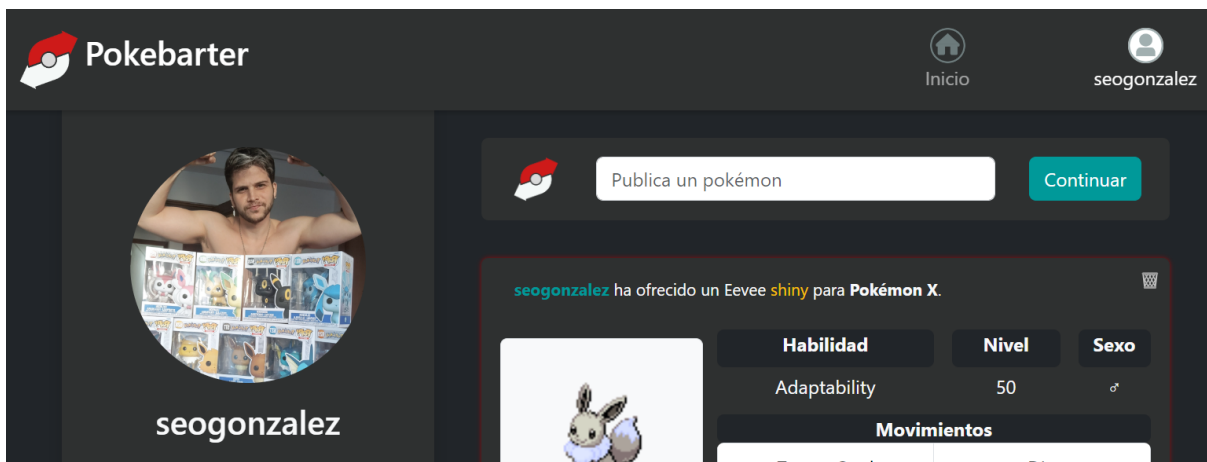


Ilustración 27 - Barra de posteo en perfil propio.

- Se accede a perfil ajeno:

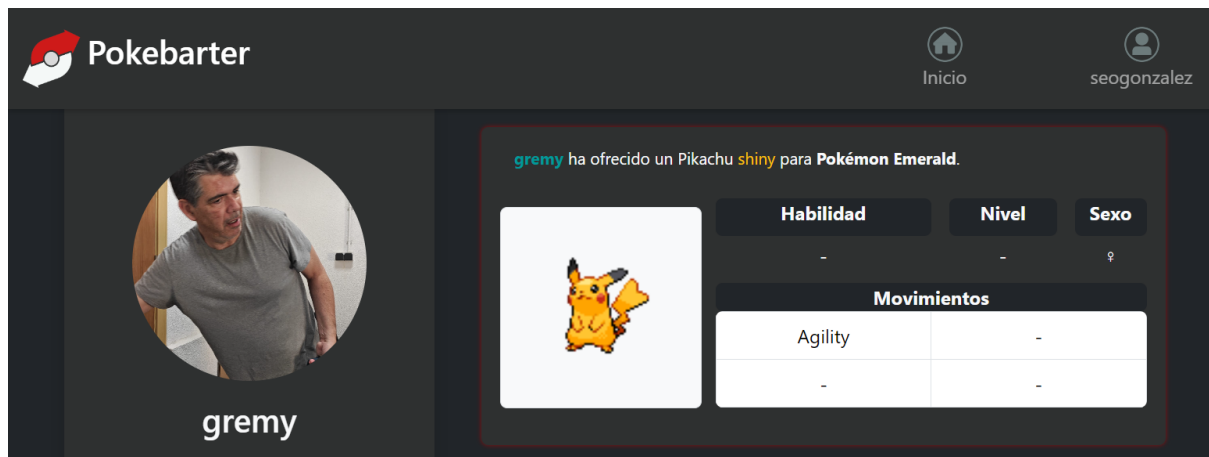


Ilustración 28 - Perfil ajeno sin barra de posteo.

- Se accede al inicio:

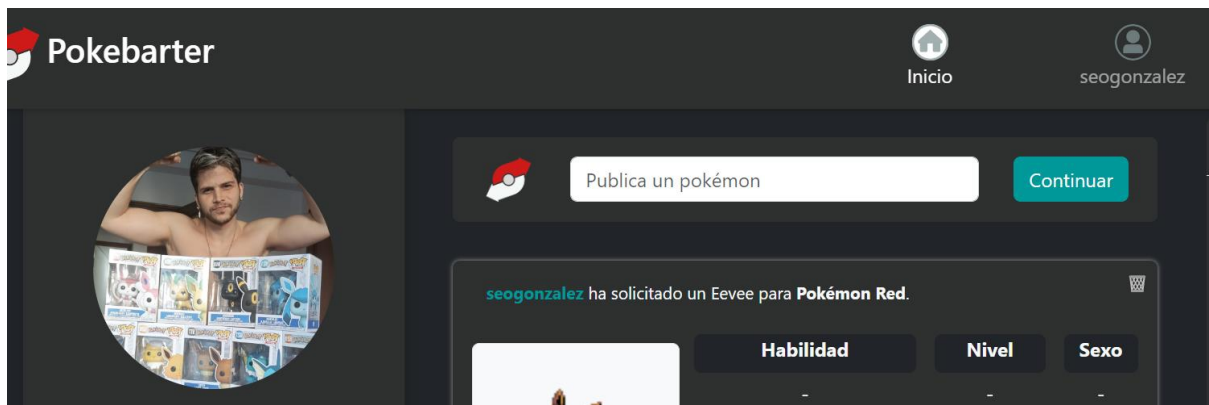


Ilustración 29 - Barra de posteo en pantalla de inicio.

En base a lo observado se puede indicar que el matriz de pruebas luciría así:

Escenario	Caso	Acción	Estado
Inicio de sesión	Se introduce un user correcto y una clave incorrecta.	No se accede y se notifica al user que los datos son incorrectos	Funciona
	Se introduce un user incorrecto y una clave correcta.	No se accede y se notifica al user que los datos son incorrectos	Funciona
	Se introduce un user correcto y una clave correcta.	Se accede	Funciona
Publicar un pokémon	Se introduce el nombre de un pokémon correcto y se presiona continuar:	No se despliega el formulario y se informe al user que el pokémon no es válido	Funciona
	Se introduce el nombre de un pokémon incorrecto y se presiona continuar	Se despliega el formulario	Funciona
Filtrado de pokémon	Se filtra por shiny	Se muestran sólo pokémon shiny	Funciona
	Se filtra por nombre de pokémon	Se muestran sólo pokémon de la misma especie que el seleccionado	Funciona
	Se filtra con estado ofrecido	Se muestran sólo pokémon que han sido ofrecidos	Funciona
	Se filtra con estado solicitado	Se muestran sólo pokémon que han sido solicitados	Funciona
Manejo de publicación	Se accede al perfil propio	Se muestra la barra para realizar publicaciones	Funciona
	Se accede a perfil ajeno	No se muestra la barra para realizar publicaciones	Funciona
	Se accede al inicio	Se muestra la barra para realizar publicaciones	Funciona

Ilustración 30 - Matriz de pruebas.

7.6.3 Pruebas de aceptación

En esta etapa de las pruebas se realiza la revisión de los requisitos funcionales, asegurándose de esta forma que la aplicación cumpla con estos, para corroborar que Pokebarter cumple con los mismos:

- Manejo de sesiones:

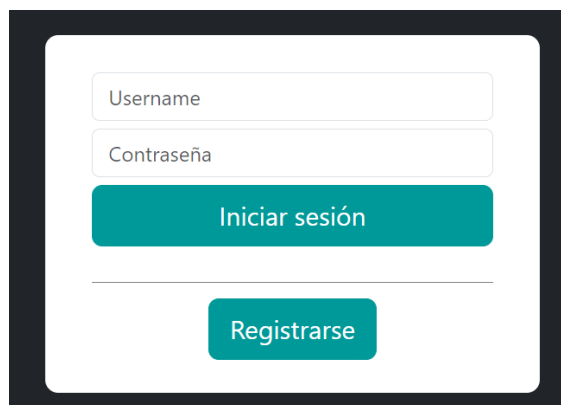
Ilustración 31 - Formulario para iniciar sesión. El formulario está centrado en una pantalla blanca con un fondo negro. Contiene dos campos de entrada: 'Username' y 'Contraseña'. Debajo de estos campos hay un botón verde con el texto 'Iniciar sesión'. En la parte inferior del formulario, separado por una línea horizontal, hay otro botón verde con el texto 'Registrarse'.

Ilustración 31 - Formulario para iniciar sesión.

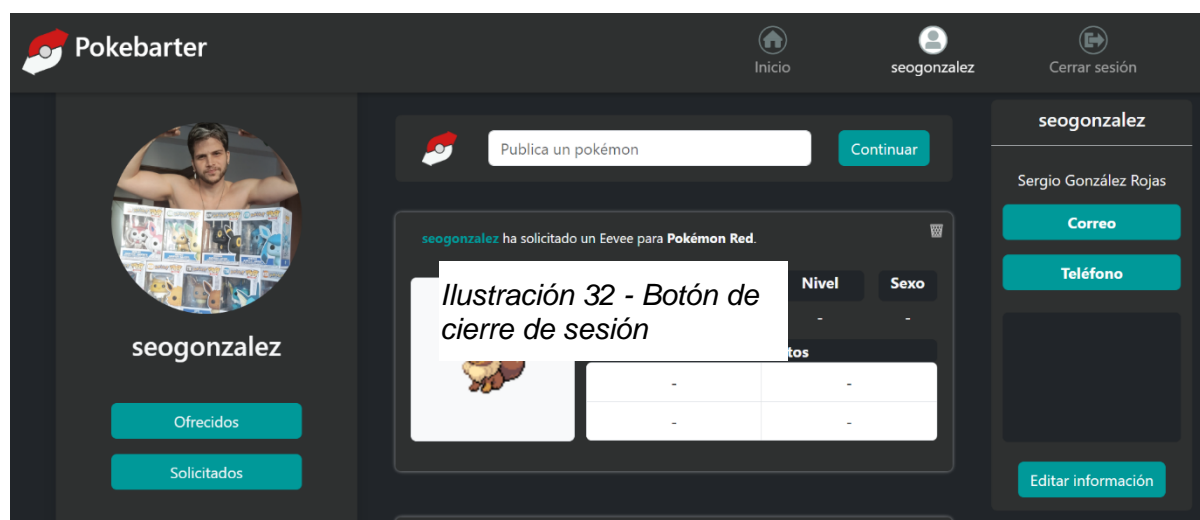


Ilustración 32 - Perfil personalizado del user.

- Creación y gestión de perfiles:



Ilustración 33 - Botón de cierre de sesión.

Sergio González Rojas

seogonzalez

sergioagrvi@gmail.com

654587900

¿Algo que quieras contar?

Contraseña*

Guardar Cancelar

Ilustración 34 - Ventana de edición de información del user.

Pokebarter

Inicio seogonzalez Cerrar sesión

gremy

Ofrecidos Solicitados

gremy ha ofrecido un Pikachu shiny para Pokémon Emerald.

Habilidad	Nivel	Sexo
-	-	?

Movimientos

Agility	-
-	-

gremy

Sergio González Rojas

Correo

Teléfono

Holiiiiin

Ilustración 35 - Perfil de user ajeno.

- Publicación de contenido:

Formulario de posteo for Eevee Gmax. The form includes a header with the Pokémon icon and name, a 'Continuar' button, a 'Movimientos' (Moves) table with two empty rows, and fields for 'Habilidad' (Ability), 'nivel' (Level), 'Sexo' (Gender), 'Shiny' (checkbox), and 'Pokémon Go' (text). At the bottom are 'Solicitar', 'Ofrecer', and 'Cancelar' buttons.

Ilustración 36 - Formulario de posteo.

Formulario de posteo showing a dropdown menu for 'Movimientos'. The menu lists various moves: Tackle, Take Down, Tail Whip, Toxic, Tickle, Trump Card, Tera Blast, and Trailblaze. The 'Shiny' checkbox is checked, and the 'Pokémon X' text is visible.

Ilustración 37 - Asistencia a los usuarios en formulario de posteo.

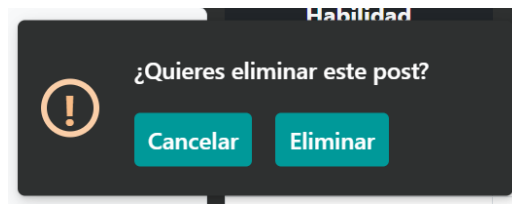


Ilustración 38 - Ventana de confirmación de anulación de post.

- Gestión de contenido de la aplicación:

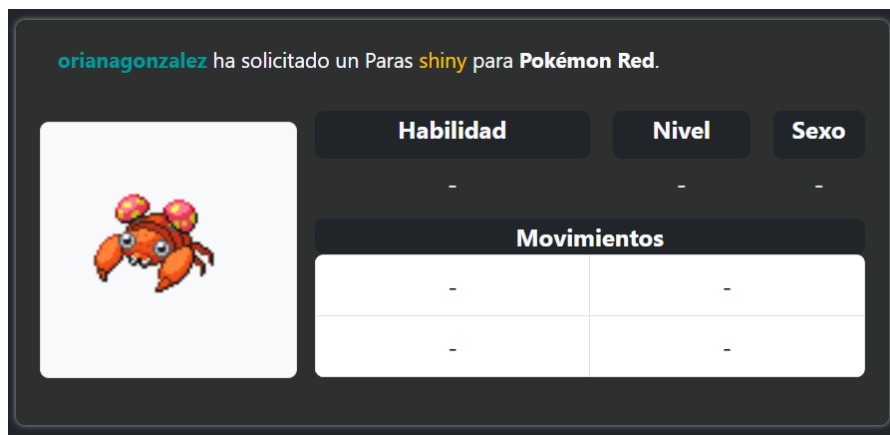


Ilustración 40 - Publicación de user ajeno.

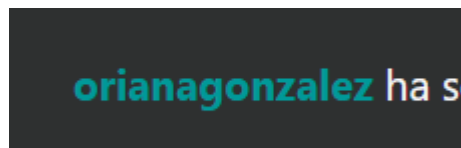


Ilustración 39 - Link de acceso a perfil del propietario de la publicación.

7.6.4 Pruebas de seguridad

Estas pruebas se centran en corroborar que el sistema sera resistente a ataques y certificar que no tenga vulnerabilidades. Existen muchos tipos de ataques, un ejemplo de este podría ser las inyecciones SQL, para prevenir que estas se puedan producir Pokebarter utiliza las consultas preparadas, en donde los parámetros se pasan como marcadores de posición, y los valores reales se adjuntan posteriormente en un array.

```
y('UPDATE user SET img = ? WHERE username = ?;', [user.img, user.username] ,  
err){
```

Ilustración 41 - Consulta preparada.

Como se pudo demostrar con anterioridad, Pokebarter también valida la funcionalidad de que los mecanismos de autenticación y autorización funcionan correctamente y son seguros.

7.7 DESPLIEGUE Y MANTENIMIENTO.

7.7.1 Despliegue

Cuando hablamos del despliegue de una aplicación web nos referimos a mover la aplicación de un entorno de desarrollo al entorno de producción, donde los users pueden acceder a ella directamente. Para desplegar Pokebarter en el entorno de gestión se decidió usar el servicio Heroku, en este caso, se usará el servicio gratuito.

Lo será instalar los programas que son necesarios para desplegar Pokebarter.

- GitHub Desktop: Permitirá clonar el repositorio para poder realizar los cambios.
- Git: Esta permitirá realizar el control de versiones y revisar su historial.

Con el repositorio de Pokebarter subido a GitHub y la sesión iniciada en el cliente de escritorio, se elige el repositorio que se desea desplegar y se clona, y una vez clonado se usa VS Code para abrir el repositorio clonado.

Una vez abierto Visual Studio Code, a continuación, se realiza la instalación de Heroku a través del terminal:

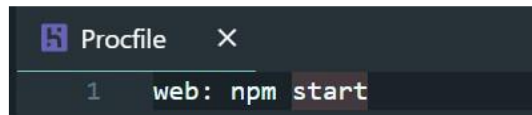
```
PS C:\Users\sergi\Desktop\Pokebarter> npm install -g heroku
```

Ilustración 42 - Comando de instalación de Heroku.

Con la sesión iniciada en el sitio web de Heroku y desde el terminal se añade un nuevo archivo a la aplicación con el nombre Procfile con la instrucción web

```
PS C:\Users\sergi\Desktop\Pokebarter> heroku login
```


Ilustración 43 - Comando de inicio de sesión de Heroku.



```
Procfile  X
1  web: npm start
```

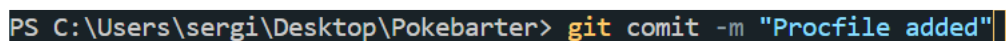
Ilustración 44 - Contenido de archivo Procfile.

Antes de implementarla se confirma el código de la aplicación en un repositorio git local en donde se indica el archivo Procfile añadido para posteriormente crear la aplicación que en primera instancia se encontraría vacía y así se creará un repositorio vacío asociado a Pokebarter, este en principio se creará como remoto para nuestro repositorio local.



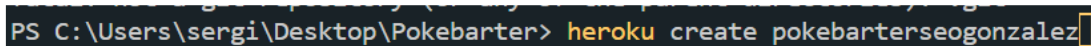
```
PS C:\Users\sergi\Desktop\Pokebarter> git add .
```

Ilustración 45 - Comando para iniciar el repositorio local.



```
PS C:\Users\sergi\Desktop\Pokebarter> git commit -m "Procfile added"
```

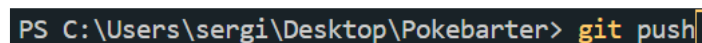
Ilustración 46 - Comando para informar al repositorio que se ha añadido el archivo Procfile.



```
PS C:\Users\sergi\Desktop\Pokebarter> heroku create pokebarterseogonzalez
```

Ilustración 47 - Comando para crear repositorio.

Después de tener todos los pasos correctamente asociados es el momento de desplegar la aplicación con git y de esta manera la app se ve desplegada.



```
PS C:\Users\sergi\Desktop\Pokebarter> git push
```

Ilustración 48 - Comando para desplegar aplicación.

7.7.2 Mantenimiento

En la fase de mantenimiento de la aplicación se garantiza que esta con el pasar del tiempo se mantenga funcionando correctamente cumpliendo siempre con las necesidades de los usuarios que van evolucionando constantemente. Pokebarte tiene un plan de mantenimiento en el que se realizaran diferentes tipos para garantizar que se mantenga siempre en producción.

- **Mantenimiento correctivo:** Se pretende en este solucionar los bugs que puedan generarse en la aplicación y los errores que los usuarios reporten en su funcionamiento.
- **Mantenimiento adaptativo:** En este se pretende llevar a Pokebarter a un nivel en el cual pueda integrarse perfectamente con las nuevas versiones de los sistemas operativos. Se espera a futuro poder integrarlo con aplicaciones de terceros, como podría ser por ejemplo X.
- **Mantenimiento perfectivo:** El principal objetivo de este en Pokebarter será garantizar que la interfaz se vea mejorada, dándole a los users en algún punto un nuevo nivel de autonomía que les permita editar perfiles con mayor libertad, elegir temas que personalicen su interfaz al gusto propio.
- **Mantenimiento preventivo:** Con la continua revisión y estudio del mercado en el que Pokebarter pretende posicionarse, se planea realizar revisiones continuas del código fuente para que este esté siempre adaptado a los cambios y se prevean posibles errores.

8. CONCLUSIONES Y MEJORAS FUTURAS

Pokémon es una franquicia enorme que ha producido a lo largo de su periodo en el mercado una variedad extensa de juegos y aplicaciones, sin embargo es cierto que a pesar de su inmensa cantidad de productos, en ciertas ocasiones se ha visto en requerimiento de verse asistido a través de otras aplicaciones, por lo cual el mercado alterno que este ha generado en su entorno ha ido creciendo, incorporando con el tipo cada vez más herramientas o productos que dependen de la marca, y que a su vez hacen de la misma más atractiva para el consumidor, puesto que facilitan el proceso de consumo.

En base a lo contemplado, se puede decir que existen entonces múltiples alternativas e ideas creadas para asistir a los productos originales de la franquicia de Pokémon, sin embargo, el desarrollo de cada uno de estos productos lleva por detrás un proceso de desarrollo arduo que comienza a complicarse desde el pensar una simple idea, hasta llevarla a cabo y desplegarla. Pokebarter es una iniciativa desarrollada para asistir a los jugadores de Pokémon en un entorno relacionado a la mecánica de intercambio, que presenta una interfaz semejante a la de una red social para así conceder una experiencia más intuitiva a los usuarios, sin a su vez perder orientación con relación a su objetivo principal.

Podemos incluir entonces a Pokebarter en este grupo de herramientas que ayudan a los jugadores a suplir esta ventana de necesidad que deja abierta la compañía de Pokémon con sus productos, la cual ha sido desde los inicios de esta fuente recurrente de inconvenientes para aquellos fans dedican su tiempo a los productos de la franquicia. En vista de que Pokémon a lo largo de su trayectoria no ha encontrado la manera de cumplimentar esta brecha de existe entre los jugadores y los juegos, los fans se deben tomar esta libertad que les permita asistirse para de esta manera poder alcanzar los objetivos que Pokémon plantea con sus entregas.

De esta manera, podemos observar que el objetivo de los fanáticos es cada vez más claro, continuarán buscando alternativas para conseguir el objetivo que Pokémon pone en frente, y además, el despliegue de este sigue motivando a la compañía a producir más y nuevas entregas, por lo tanto se podría llegar a la conclusión de que este sistema cíclico en el que participan tanto los consumidores de los productos como la misma compañía de Pokémon genera una retroalimentación que favorece a ambas partes, sin embargo el camino es aún largo para lograr un entorno ideal en la comunidad de Pokémon, puesto que según pasa el tiempo, mecánicas siguen surgiendo y siempre será indispensable seguir manteniendo a la comunidad comunicada entre sí para dar una experiencia que sobrepase las expectativas.

Visto de esta manera, Pokebarter es una iniciativa sencilla con objetivos claros y directos, sin embargo, el camino aún es largo puesto que con el constante crecimiento de usuarios la implementación de nuevas funcionalidades será requerida. La inclusión de sistemas de amistad es una actividad que se prevé desde el primer momento, que podría darles acceso a los usuarios sólo a publicaciones de aquellos que se encuentren entre sus amigos, planteando a Pokebarter en los marcos de las redes sociales.

Además, el sistema de gestión de la aplicación puede ser utilizado en ámbitos diferentes a los de la idea original, alcanzando entornos diferentes a los de Pokémon, como podrían ser todas aquellas franquicias que se planten en el interés por coleccionismo, ejemplos tales como pueden ser el exitoso Dragon Quest, Monstres Hunter o incluso la competencia directa de Pokémon, Digimon.

9. BIOGRAFÍA Y REFERENCIAS

Mozilla y Contribuidores de la Comunidad. Mozilla Developer Network Web Docs.

<https://developer.mozilla.org/es/docs/Web/HTML>,

<https://developer.mozilla.org/es/docs/Web/CSS>,

<https://developer.mozilla.org/es/docs/Web/JavaScript>

PokéAPI. Documentation – PokéAPI. <https://pokeapi.co/docs/v2>

Antonio José Peleteiro Crespo. (2021, 08 de enero). Bottempo.

<https://openaccess.uoc.edu/bitstream/10609/126529/8/apeleteiroTFG0121memoria.pdf>

Facebook y Contribuyentes de la Comunidad. Delightful JavaScript Testing.

<https://jestjs.io/es-ES/>

Bootstrap. Bootstrap. <https://getbootstrap.com/>

Node.js. Node.js — Run JavaScript Everywhere. <https://nodejs.org>

Express.js. Express - Node.js web application framework. <https://expressjs.com>

10. ANEXOS

GLOSARIO DE TÉRMINOS

Aplicación Web: Es un programa que se ejecuta en un servidor web y se accede a través de un navegador web. A diferencia del software de escritorio que se instala localmente en un sistema operativo, las aplicaciones web están disponibles en línea y pueden ser utilizadas desde cualquier dispositivo con acceso a internet.

Back-end: Es la parte de una aplicación que maneja la lógica del servidor, la base de datos, y la comunicación con el front-end. Procesa las solicitudes de los usuarios, realiza operaciones con la base de datos, y devuelve la información al front-end.

Servidor Web: Es un software y hardware que utiliza HTTP y otros protocolos para responder a las solicitudes de los usuarios que llegan a través de la red. Su principal función es almacenar, procesar y entregar páginas web a los usuarios. Los servidores web pueden gestionar peticiones tanto estáticas como dinámicas y servir contenidos como HTML, imágenes, vídeos, o aplicaciones web.

Base de Datos: Es un sistema organizado para almacenar, gestionar y recuperar datos. Permite a los usuarios y aplicaciones crear, leer, actualizar y eliminar datos de manera eficiente.

Front-end: Es la parte de una aplicación que interactúa directamente con el usuario. Incluye todo el diseño visual, la estructura, el comportamiento y la interactividad de una página web o aplicación. Está desarrollado usando tecnologías como HTML, CSS y JavaScript.

Navegador Web: Es una aplicación de software que permite a los usuarios acceder, recuperar y visualizar información en la World Wide Web. Los navegadores interpretan el código HTML de las páginas web para mostrar contenido visual en pantalla, y pueden gestionar otros tipos de archivos como imágenes, vídeos, y archivos descargables.

Interfaz de Usuario / UI: Es la parte de una aplicación o sistema que interactúa directamente con el usuario. Incluye todos los elementos visuales, gráficos, y componentes interactivos como botones, menús, y formularios que permiten al usuario interactuar con el sistema.

Script: Es un conjunto de instrucciones que se ejecutan automáticamente en lugar de ser compiladas. Los scripts se utilizan para automatizar tareas, configurar sistemas, o ejecutar programas más complejos.

Código Fuente: Es un conjunto de instrucciones escritas en un lenguaje de programación que define el comportamiento y la funcionalidad de un software o aplicación.

Aplicación Escalable: Es aquella aplicación que es capaz de manejar un aumento en la carga de trabajo o la cantidad de usuarios sin degradar su rendimiento.

Archivo: Es una colección de datos o información que se guarda en una unidad de almacenamiento bajo un nombre particular.

Pokémon: Es una criatura ficticia en el universo Pokémon, que los jugadores pueden capturar, entrenar y utilizar en batallas. Cada Pokémon tiene diferentes características, tipos y habilidades.

Pokémon Shiny: Es una versión extremadamente rara de un Pokémon que tiene una coloración diferente a la normal. La probabilidad de encontrar un Pokémon shiny es muy baja, y suelen ser buscados por coleccionistas.

Habilidades de un pokémon: Son características especiales que pueden influir en las batallas. Cada Pokémon puede tener una o varias habilidades que afectan su comportamiento, sus movimientos o sus estadísticas en combate.

IMÁGENES DE LA VERSIÓN MÓVIL

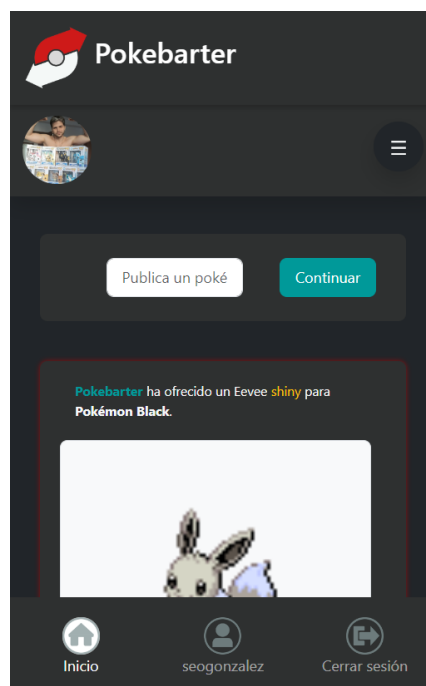


Ilustración 49 - Pantalla de inicio en el formato móvil

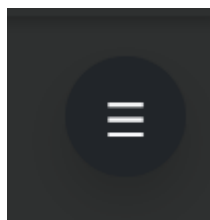


Ilustración 50 - Botón de despliegue de laterales.

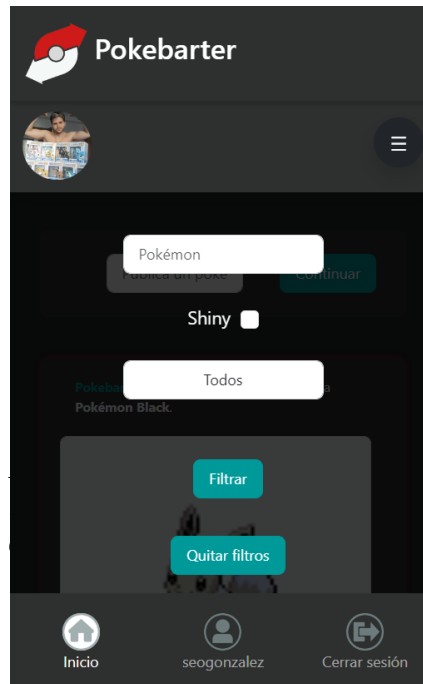


Ilustración 51 - Zona de filtrado desplegada en el formato móvil.

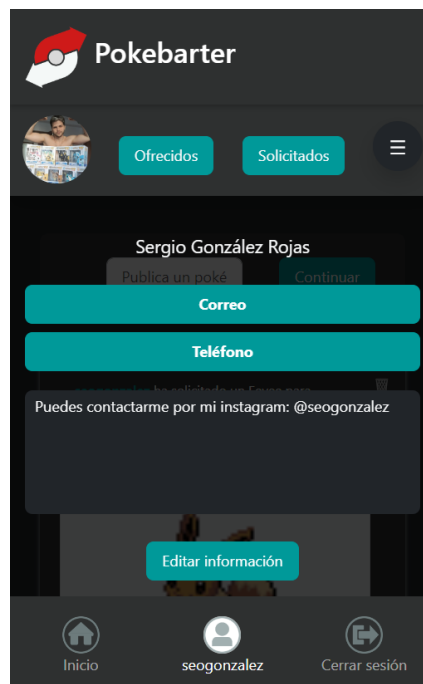


Ilustración 52 - Zona de información personal desplegada en el formato móvil.