Gráfico de burbujas

Descripción generada automáticamente con confianza baja

**ÍNDICE**

[DESCRIPCIÓN GENERAL DEL PROYECTO 1](#_Toc136806732)

[ESTUDIO DE VIABILIDAD DEL SISTEMA 2](#_Toc136806733)

[ANÁLISIS DAFO 2](#_Toc136806734)

[PLAN DE MARKETING 4](#_Toc136806735)

[OBJETIVOS Y AUDIENCIA 4](#_Toc136806736)

[PROPUESTA DE VALOR 4](#_Toc136806737)

[ESTRATEGIA DE CONTENIDOS 4](#_Toc136806738)

[ESTRATEGIAS POST-LANZAMIENTO 5](#_Toc136806739)

[REQUISITOS FUNCIONALES y NO FUNCIONALES 5](#_Toc136806740)

[REQUISITOS FUNCIONALES 5](#_Toc136806741)

[REQUISITOS NO FUNCIONALES 6](#_Toc136806742)

[CASOS DE USO 6](#_Toc136806743)

[MODELO DE CASOS DE USO 6](#_Toc136806744)

[CASOS DE USO 7](#_Toc136806745)

[DISEÑO E/R 15](#_Toc136806746)

[NORMALIZACIÓN DE TABLAS 16](#_Toc136806747)

[DIAGRAMA DE CLASE 17](#_Toc136806748)

[VERSIÓN 0.0.1 17](#_Toc136806749)

[VERSIÓN 0.1 18](#_Toc136806750)

[VERSIÓN 1.0 19](#_Toc136806751)

[DISEÑO DE INTERFAZ 20](#_Toc136806752)

[TABLA DE IMÁGENES 20](#_Toc136806753)

[GUÍA DE ESTILOS 28](#_Toc136806754)

[DISEÑO DE PRUEBAS 29](#_Toc136806755)

# DESCRIPCIÓN GENERAL DEL PROYECTO

El objetivo inicial del proyecto consiste en crear una aplicación web que facilitará las necesidades del creador a la hora de compartir archivos entre equipos. Esta iniciativa surge con la dificultad de su creador a la hora de manejar y entender plataformas como Github, que se apoyan en software como Git. **Repositor.io** se transformó poco a poco en una plataforma de almacenamiento en la nube gratuita para todos sus usuarios.

El proyecto se ha desarrollado para uso exclusivo en equipos de escritorio, contando crear una futura expansión a dispositivos móviles. Para su desarrollo se han utilizado las siguientes tecnologías:

* Interfaces diseñadas en Angular
* API REST creada con .NET Core
* Base de datos en MS SQL SERVER

La aplicación va dirigida a todo usuario que desee compartir archivos sin preocuparse por planes premium, tamaños limitados de subida u otras inconveniencias como pueden importunar software como Git con Github o Google Drive.

# ESTUDIO DE VIABILIDAD DEL SISTEMA

## ANÁLISIS DAFO

|  |  |
| --- | --- |
| **ANÁLISIS DAFO** | |
| **ANÁLISIS INTERNO** | **ANÁLISIS EXTERNO** |
| **DEBILIDADES** | **AMENAZAS** |
| * **Competencia.** El mercado de almacenamiento de archivos en línea es altamente competitivo, con grandes actores como Google Drive, Dropbox, OneDrive, entre otros. Sería necesario asegurarse de que el producto pudiese ofrecer características únicas, o destacar en algún aspecto, con el objetivo de diferenciarse del resto de la oferta. * **Capacidad de almacenamiento.** Sería conveniente evaluar y asegurar que la infraestructura sea capaz de manejar el almacenamiento y la gestión de los archivos de manera eficiente y escalable. | * **Seguridad.** La seguridad y privacidad de los archivos de los usuarios son preocupaciones importantes. Se debería garantizar la implementación de medidas de seguridad sólidas para proteger la información de los usuarios y evitar posibles brechas de seguridad. * **Cambios en el mercado.** El mercado de almacenamiento en línea está en constante evolución, y podrían surgir nuevas tecnologías o cambios en las preferencias de los usuarios que podrían afectar la demanda de la aplicación. |
| **FORTALEZAS** | **OPORTUNIDADES** |
| * **Simplicidad.** Al ser una versión distinta, en cierto sentido más directa y sencilla que Google Drive, la aplicación puede atraer a usuarios que buscan una experiencia más intuitiva y fácil de usar. * **Almacenamiento de archivos**. La capacidad de almacenar y organizar archivos en repositorios proporcionará a los usuarios un lugar centralizado para gestionar sus documentos digitales. * **Acceso desde cualquier lugar.** Al ser una aplicación web, los usuarios podrán acceder a sus archivos desde cualquier dispositivo con conexión a internet. | * **Necesidad de simplicidad**. Existe una demanda creciente de herramientas y aplicaciones más sencillas y fáciles de usar, lo que podría beneficiar a la aplicación una vez en el mercado. * **Personalización**. Se podrían agregar funciones que habilitasen la personalización para permitir a los usuarios adaptar la apariencia de sus repositorios, dándole esto un carácter más individual a la experiencia de los usuarios con la aplicación web, todo ello acorde con sus necesidades. |

## PLAN DE MARKETING

### OBJETIVOS Y AUDIENCIA

Los objetivos de **Repositor.io** como empresa sin ánimo de lucro serán puramente expansivos. Se tratará de alcanzar un número de descargas elevado (1000) en el primer semestre tras el lanzamiento de la aplicación al mercado. Asimismo, se buscará aumentar el conocimiento de la marca y captar donaciones. Al ser una iniciativa sin ánimo de lucro, los servicios de **Repositor.io** tendrán como objetivo instituciones públicas, individuos autónomos, etc.

En comparación con otras plataformas similares, **Repositor.io** elimina la complejidad del proceso de almacenamiento en nube.

### PROPUESTA DE VALOR

La ventaja de **Repositor.io** frente a otros servicios similares será la sencillez y facilidad de uso, eliminando funcionalidades que pueden llegar a abrumar al usuario en procesos relativamente sencillos como son guardar y descargar archivos, fomentando un uso continuo para sus usuarios.

Como ejemplo, esta iniciativa pretende eliminar el uso de memorias USB para los sectores no-tecnológicos, siendo estos los que más tardan en actualizarse con tecnologías punteras como servicios en la nube. Se trata de una propuesta similar a la implementación de *Microsoft Teams* en la educación de casi todo el mundo durante la pandemia de 2020.

### ESTRATEGIA DE CONTENIDOS

Se buscará publicar en redes contenidos informativos y de carácter publicitario para aumentar la ratio de *engagement* del usuario. La aplicación web implementará estrategias SEO para fomentar el índice de aparición en motores de búsqueda indexados como Google, además de una estructura que aproveche diseños y elementos Material, fomentado por Google.

### ESTRATEGIAS POST-LANZAMIENTO

Se monitorizará el uso de la aplicación web para tratar de refinar y ajustar los aspectos relevantes tras su lanzamiento. Al mismo tiempo, se establecerá un canal de comunicación directa con los usuarios, permitiéndoles dejar comentarios y plantear dudas al responsable elegido.

# REQUISITOS FUNCIONALES y NO FUNCIONALES

## REQUISITOS FUNCIONALES

* Los usuarios anónimos sólo podrán acceder a las áreas comunes.
* Los usuarios sólo podrán registrar repositorios si están registrados dentro de la aplicación web.
* Los usuarios sólo podrán registrar repositorios si están autenticados.
* Para registrar una cuenta, el usuario deberá proporcionar:
  + Nombre de usuario
  + Correo electrónico
  + Contraseña
* Para identificarse y acceder a su cuenta, el usuario deberá proporcionar las siguientes credenciales:
  + Nombre de usuario
  + Contraseña
* Para poder registrar un nuevo repositorio a nombre de un usuario, se deberán rellenar los siguientes campos:
  + Nombre
  + Descripción
* Los usuarios autenticados podrán buscar y seguir a otros usuarios.
* El sistema permitirá visualizar los archivos de cada repositorio que albergue su cuenta de usuario.

## REQUISITOS NO FUNCIONALES

* Los datos de la aplicación quedarán almacenados en una base de datos, gestionándose a través de un sistema gestor de bases de datos.
* Las peticiones de seguimiento aceptadas entre usuarios permitirán a estos acceder a los repositorios que guarden, pudiendo el usuario que da acceso adaptar los permisos del usuario externo para modificar y/o descargar contenido de los repositorios.
* Los posibles roles de las cuentas de usuario serán:
  + ADMINISTRADOR
  + USUARIO
* Los tiempos de carga no deben ser elevados.
* El acceso a datos estará asegurado.

# CASOS DE USO

## MODELO DE CASOS DE USO

## CASOS DE USO

|  |  |  |
| --- | --- | --- |
| **ID** | **CU001** | |
| **NOMBRE** | **Registro de cuenta** | |
| **DESCRIPCIÓN** | Registrar una cuenta de usuario nueva en el sistema. | |
| **PRECONDICIÓN** | Los datos del usuario no deben existir en el sistema previo a su registro. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario hace click en el enlace de registro en la barra de navegación. |
| **2** | El servidor recibe la petición del usuario y envía el formulario de registro. |
| **3** | El usuario rellena los datos del formulario. Pueden darse dos casos, a saber:   * CU001\_FP3\_DATOS\_CORRECTOS * CU001\_FP3\_DATOS\_INCORRECTOS   Seguimos el primer caso. |
| **4** | El servidor recibe los datos del usuario y los comprueba. Son correctos y los guarda en el sistema. Devuelve al usuario a la página de inicio de usuario. |
| **FLUJO DE ERROR** | **1** | El usuario rellena los datos del formulario. Pueden darse dos casos. Seguimos el caso CU001\_FP3\_DATOS\_INCORRECTOS. |
| **2** | El servidor recibe los datos del usuario y los comprueba.  Son erróneos. Devuelve al usuario a la misma página, mostrando un menú modal en el que comunica el error y una descripción |
| **3** | Al cerrar el menú, se volverá a mostrar el formulario de registro. |
| **POSTCONDICIÓN** | El sistema redirige al usuario a la página de inicio de usuario. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU002** | |
| **NOMBRE** | **Inicio de sesión** | |
| **DESCRIPCIÓN** | Iniciar sesión en la plataforma web. Se puede escoger entre el rol de administrador o el de usuario. | |
| **PRECONDICIÓN** | Registrar una cuenta de usuario. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario hace click en el enlace de la barra de navegación. |
| **2** | El servidor recibe la petición y devuelve el formulario de login al usuario. |
| **3** | El usuario rellena los datos de login y los envía. Pueden darse dos casos, a saber:   * CU002\_FP3\_DATOS\_CORRECTOS * CU002\_FP3\_DATOS\_INCORRECTOS   Seguimos el primer caso. |
| **4a** | El servidor recibe los datos del usuario y los comprueba. Son correctos y el servidor devuelve la página de inicio del usuario, con funciones intrínsecas de *usuario*. |
| **4b** | El servidor recibe los datos del usuario y los comprueba. Son correctos y el servidor devuelve la página de inicio del administrador, con funciones intrínsecas de *administrador*. |
| **FLUJO DE ERROR** | **1** | El usuario rellena los datos de login y los envía. Seguimos el caso CU002\_FP3\_DATOS\_INCORRECTOS |
| **2** | El servidor recibe los datos y los comprueba.  Son erróneos y el servidor devuelve la misma página de login mostrando un menú modal en el que comunica el error y una descripción. |
| **3** | Al cerrar el menú modal, se mostrará de nuevo el formulario de login. |
| **POSTCONDICIÓN** | El sistema redirige al usuario/administrador a la página de inicio respectiva. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU003** | |
| **NOMBRE** | **Búsqueda de usuarios** | |
| **DESCRIPCIÓN** | Buscar usuarios registrados en la base de datos. | |
| **PRECONDICIÓN** | Haber iniciado sesión para obtener los derechos de esta acción. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario introduce un término de búsqueda en la barra de búsqueda. |
| **2** | El servidor recibe la petición y devuelve el listado con los resultados de la búsqueda. |
| **3** | Pueden darse dos casos:   * CU003\_FP3\_USUARIO\_EXISTE * CU003\_FP3\_ USUARIO\_NO\_EXISTE |
| **4** | Siguiendo el flujo normal del programa, se mostrará un listado de nombres de usuario que coincidan con el término de búsqueda del usuario. |
| **FLUJO DE ERROR** | **1** | El usuario ha buscado un término que no aparece en la base de datos, con lo que el servidor devolverá un mensaje de fallo en la búsqueda. |
| **POSTCONDICIÓN** | El sistema mostrará correctamente los resultados de la búsqueda del usuario. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU004** | |
| **NOMBRE** | **Seguir usuario** | |
| **DESCRIPCIÓN** | Añadir seguimiento a la actividad de un usuario, permitiendo recibir notificaciones cuando este suba nuevo contenido a la plataforma. | |
| **PRECONDICIÓN** | Haber iniciado sesión para obtener los derechos de esta acción. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario entra en el perfil que desee seguir y pulsa el botón de “Seguir usuario”. |
| **2** | El sistema recibe la acción del primer usuario y devuelve una actualización. |
| **3** | Pueden darse dos casos:   * CU003\_FP3\_SOLICITUD\_COMPLETADA * CU003\_FP3\_ SOLICITUD\_NO\_COMPLETADA |
| **4** | Siguiendo el flujo normal del programa, se mostrará una notificación al usuario que ha pulsado en “seguir” informándole de que a partir de ahora recibirá notificaciones cada vez que este usuario suba contenido nuevo. |
| **FLUJO DE ERROR** | **1** | El usuario ha tratado de seguir a un usuario, pero el sistema ha fallado en la transmisión de la acción y devuelve un código de error ERR\_004: “Error interno”. |
| **POSTCONDICIÓN** | El usuario podrá ver actualizaciones del contenido en su página de inicio personal de todos los usuarios que siga. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU005** | |
| **NOMBRE** | **Editar perfil de usuario** | |
| **DESCRIPCIÓN** | Ser capaz de modificar la información que aparece en el perfil de cada usuario, respectivamente. | |
| **PRECONDICIÓN** | Haber iniciado sesión para obtener los derechos de esta acción. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario hace clic en el icono de su imagen de perfil, en la barra de navegación. |
| **2** | El servidor recibe la petición y devuelve la información de ese usuario. |
| **3** | Se modifica alguno de las siguientes propiedades del usuario:   * Descripción * Imagen de perfil * Proyectos   En el caso de que se produzca un fallo en el proceso, se dará uno de los siguientes flujos de error:   * CU\_005\_FP3\_MAX\_DESC\_EXCEDIDA * CU\_005\_FP3\_MAX\_SIZE\_EXCEDIDA * \*En caso de que se produzca un error al gestionar los proyectos del usuario, sea **editar** el contenido de un proyecto existente, **descargar** el proyecto entero del servidor o **eliminarlo**, se mostrará el siguiente flujo de error: CU\_005\_FP3\_PROJECT\_EXEC\_ERROR |
| **4a** | La descripción del usuario se modifica con éxito. El servidor recibe la nueva descripción. |
| **4b** | La imagen del usuario se modifica con éxito. El servidor recibe la nueva imagen de perfil. |
| **4c** | Los proyectos son modificados con éxito. El servidor recibe el nuevo estado de los proyectos del usuario. |
| **FLUJO DE ERROR** | **1** | **CU\_005\_FP3\_MAX\_DESC\_EXCEDIDA**: El usuario ha excedido el límite de caracteres válidos para la descripción. Se instará al usuario a volver a modificar la descripción para que corresponda con el límite. |
| **2** | **CU\_005\_FP3\_MAX\_SIZE\_EXCEDIDA**: El usuario ha excedido la capacidad máxima de imagen al tratar de subir una imagen muy pesada al servidor. Se instará al usuario a subir una imagen que cumpla el límite de tamaño. |
| **3** | **CU\_005\_FP3\_PROJECT\_EXEC\_ERROR**: Se ha producido un error en la transferencia de archivos, o bien otro fallo en la conexión al servidor a la hora de realizar la subida. Estos fallos incluyen tratar de subir archivos demasiado pesados. |
| **POSTCONDICIÓN** | El sistema mostrará correctamente el perfil del usuario tras las modificaciones. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU006** | |
| **NOMBRE** | **Administrar usuarios** | |
| **DESCRIPCIÓN** | Modificar el acceso de los usuarios, además de la capacidad de modificación de los repositorios de usuarios. | |
| **PRECONDICIÓN** | Que exista por lo menos un usuario con un repositorio en la base de datos. | |
| **FLUJO DE PROGRAMA** | **1** | El administrador procede a actualizar el rol del usuario *user01* a administrador del sistema. |
| **2** | La petición se envía al servidor. Este la recibe y procesa la solicitud. |
| **3** | Se acepta, de modo que devuelve una respuesta al cliente. El cliente recibe la nueva información actualizada de *user01*, que ahora es administrador. |
| **POSTCONDICIÓN** | La nueva información modificada del usuario se guardará en el sistema y sus permisos se actualizarán. | |

|  |  |  |
| --- | --- | --- |
| **ID** | **CU007** | |
| **NOMBRE** | **Subir proyecto** | |
| **DESCRIPCIÓN** | Enviar al servidor una copia de los archivos de un proyecto, subiéndolos directamente desde formulario web. | |
| **PRECONDICIÓN** | Estar registrado como usuario y haber iniciado sesión como tal. | |
| **FLUJO DE PROGRAMA** | **1** | El usuario selecciona la opción de “Subir proyecto”. |
| **2** | El servidor devuelve un formulario HTML donde se podrán volcar los archivos deseados. Tras ello, se envían al servidor. |
| **3** | El servidor los recibe y los evalúa. En esto punto pueden surgir varios caminos:   * CU007\_FP3\_FILE\_2BIG * CU007\_FP3\_ NO\_FILES * CU007\_FP3\_UP\_200 |
| **4** | En el flujo normal del ciclo, seguimos el último caso, donde los archivos se reciben en el servidor y se guardan. |
| **FLUJO DE ERROR** | **1** | **CU007\_FP3\_FILE\_2BIG**: En este caso, el tamaño total de los archivos o el de alguno en concreto excede el máximo tolerado en la subida. Este error se comunica al usuario mediante un pop-up. |
| **2** | **CU007\_FP3\_ NO\_FILES**: En este caso, el usuario ha intentado enviar el formulario sin haber volcado archivos en él. Este error se comunicará al usuario mediante otro pop-up. |
| **POSTCONDICIÓN** | El usuario recibirá una vista de los archivos que ha subido en el proyecto que ha creados. En la lista de archivos se verá la información correspondiente a cada uno. | |

# DISEÑO E/R

El diagrama entidad/relación quedará compuesto por las siguientes entidades:

* Usuario
* Cuenta de usuario
* Perfil de usuario
* Relaciones
* Repositorio
* Archivos

La entidad *Usuario* consta de un campo identificador auto incremental, nombre de usuario, correo electrónico y contraseña. Estos últimos son de tipo VARCHAR.

La entidad *Cuenta de usuario* se compone de un único *Usuario*, de modo que cada usuario puede tener una única cuenta de usuario (contando con que el usuario no use más de un email). Esto resulta en que un usuario pueda tener muchas cuentas de usuario. Además del nombre de usuario y correo electrónico, esta entidad alberga una propiedad que determina el espacio libre que le queda a ese usuario, puesto que el servidor en el que se albergarán los datos de la aplicación tendrá un límite concreto.

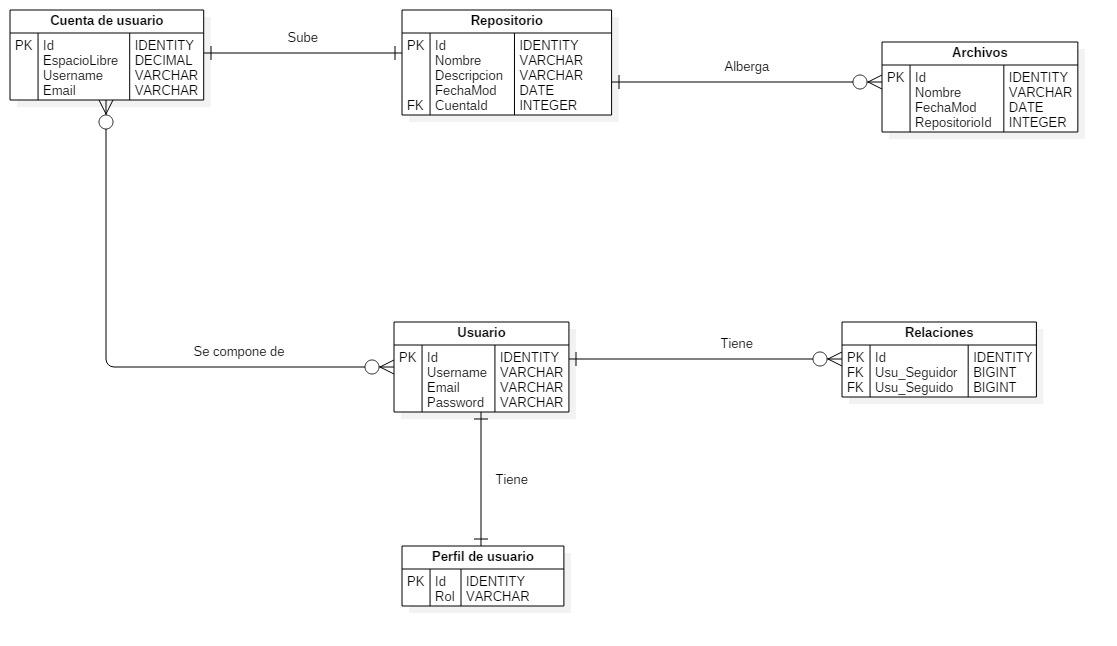
La entidad *Perfil de usuario* es una propiedad única de la entidad *Usuario*. Dentro de esta entidad se alberga el nivel de permisos del usuario. Este puede variar entre ADMINISTRADOR y USUARIO.

La entidad *Relaciones*  se trata de una tabla intermedia entre un usuario y otro. El sistema social de la aplicación consiste en un formato seguidor-seguido. Un usuario puede seguir a muchos usuarios, pero todos ellos usuarios no necesariamente seguirán al primero. Cuenta con el Id del primer usuario y el Id del segundo usuario.

La entidad *Repositorio* consta de Id, Nombre, Descripción, Fecha de modificación y una clave foránea que referencia el Id de la entidad *Cuenta de usuario*, a su vez referenciando el *Usuario* correspondiente.

La entidad *Archivos* referencia a los archivos que contendrá cada repositorio. Cada repositorio tendrá asociada una serie de filas en la entidad *Archivos*, de manera que cada una de ellas referenciará el repositorio al que pertenece, así como el usuario que subió tales archivos y una fecha de modificación.

## NORMALIZACIÓN DE TABLAS



En la imagen superior podemos ver cómo ha sido el resultado de pasar del modelo entidad/relación al modelo relacional. Para ello hemos utilizado MS SQL SERVER y los pasos que hemos llevado a cabo han sido los siguientes:

* Todas las entidades que teníamos las hemos transformado en tablas.
* Todos los atributos han sido transformados en columnas dentro de la tabla a la que pertenecen.
* Todas las id propias que hemos creado para identificar mejor cada tabla han sido puestas como claves primarias.
* Todas las id propias que hemos creado para identificar mejor cada tabla han sido puestas como claves primarias.
* En las relaciones 1: N la clave primaria de la entidad con cardinalidad 1 pasa a la otra tabla como clave foránea.

El conjunto de tablas resultante se encuentra en tercera Forma Normal (3FN). En primer lugar, se normalizó a 1FN para ello seguí los siguientes pasos:

* Suprimir la redundancia de datos.
* Proteger la integridad de los datos.
* No tener problemas a la hora de actualizar datos en las tablas.

Una vez teniendo la base de datos en 1FN pasé a 2FN de la siguiente manera:

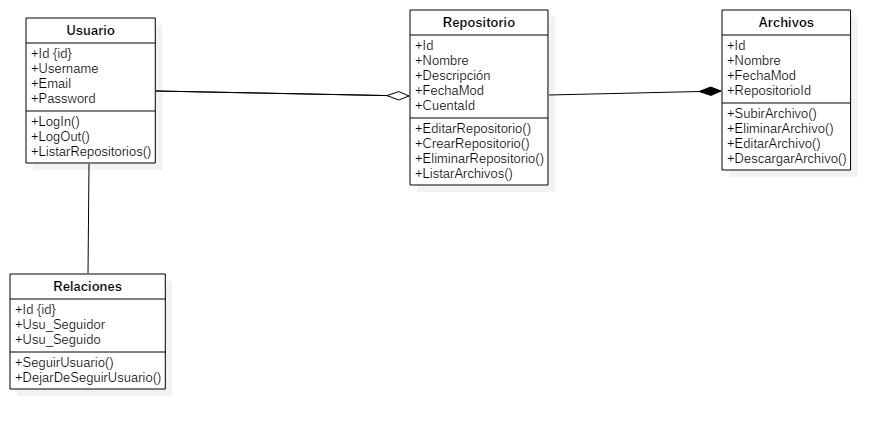
* Creando tablas separadas para todos aquellos datos con diferentes registros.
* Una clave principal externa de una única columna.

Por último, la pasé a 3FN:

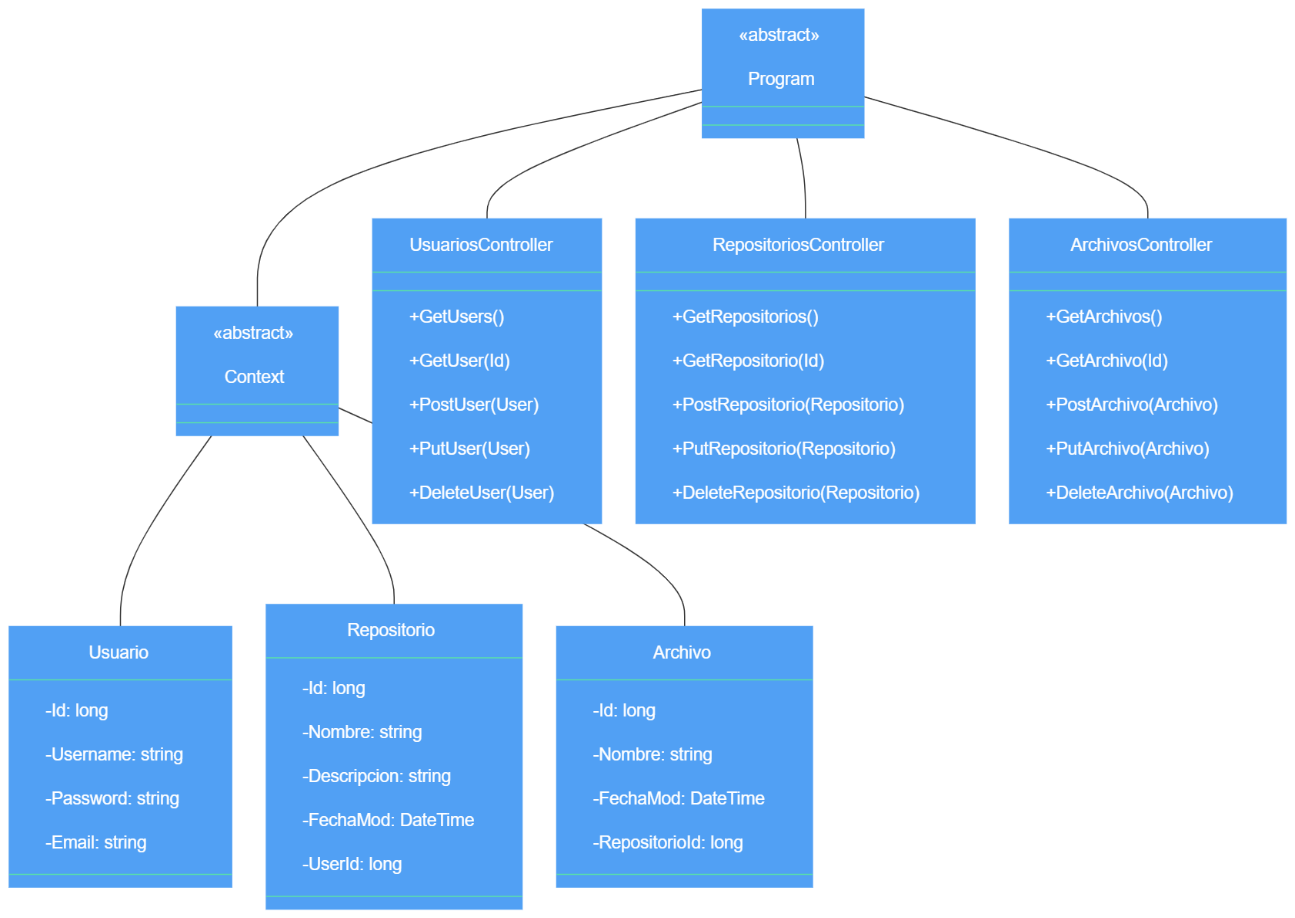
* No habiendo ninguna dependencia entre los atributos que no son clave.
* Eliminando aquellos campos que no dependan de la clave.

# DIAGRAMA DE CLASE

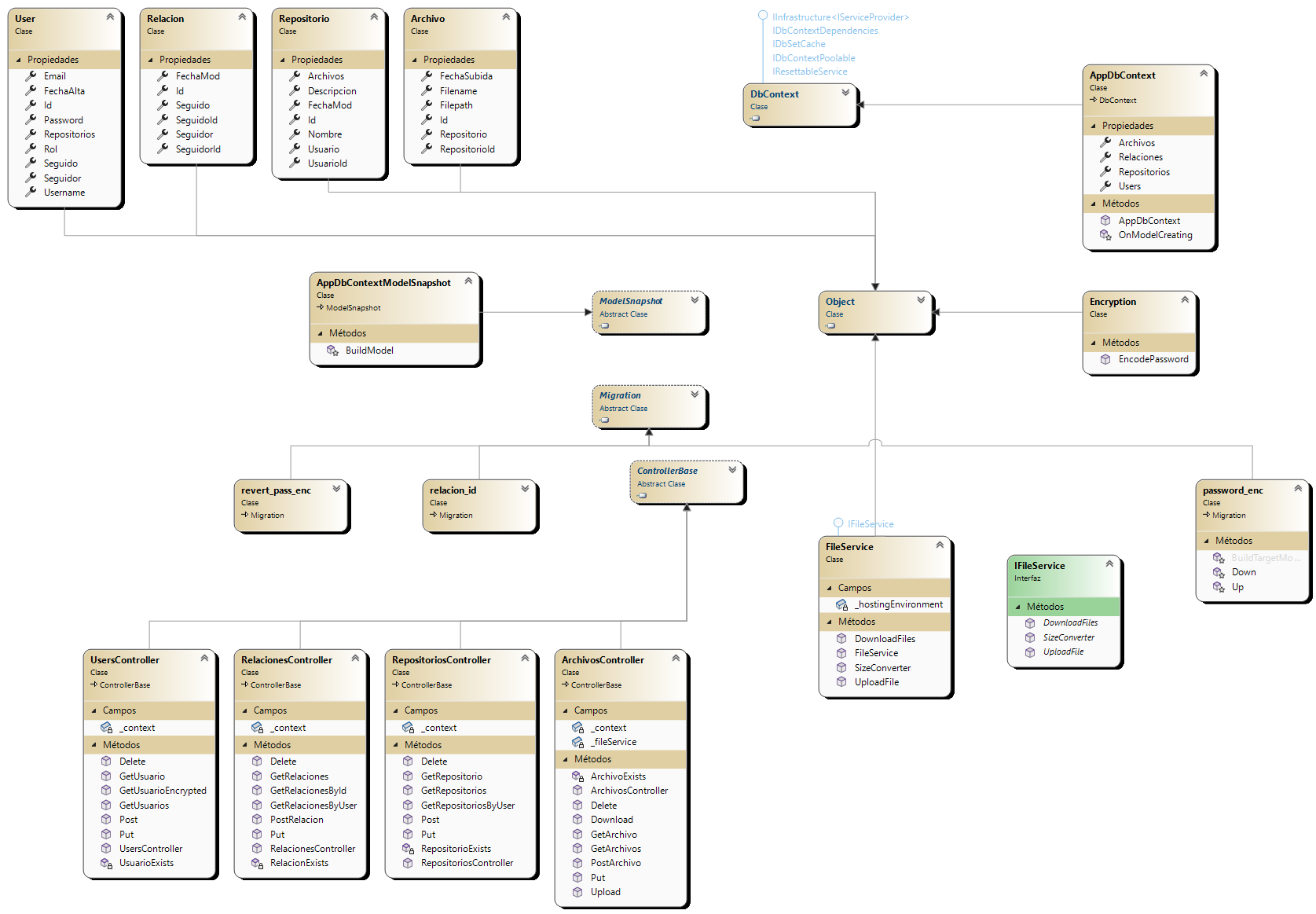
## VERSIÓN 0.0.1



## VERSIÓN 0.1

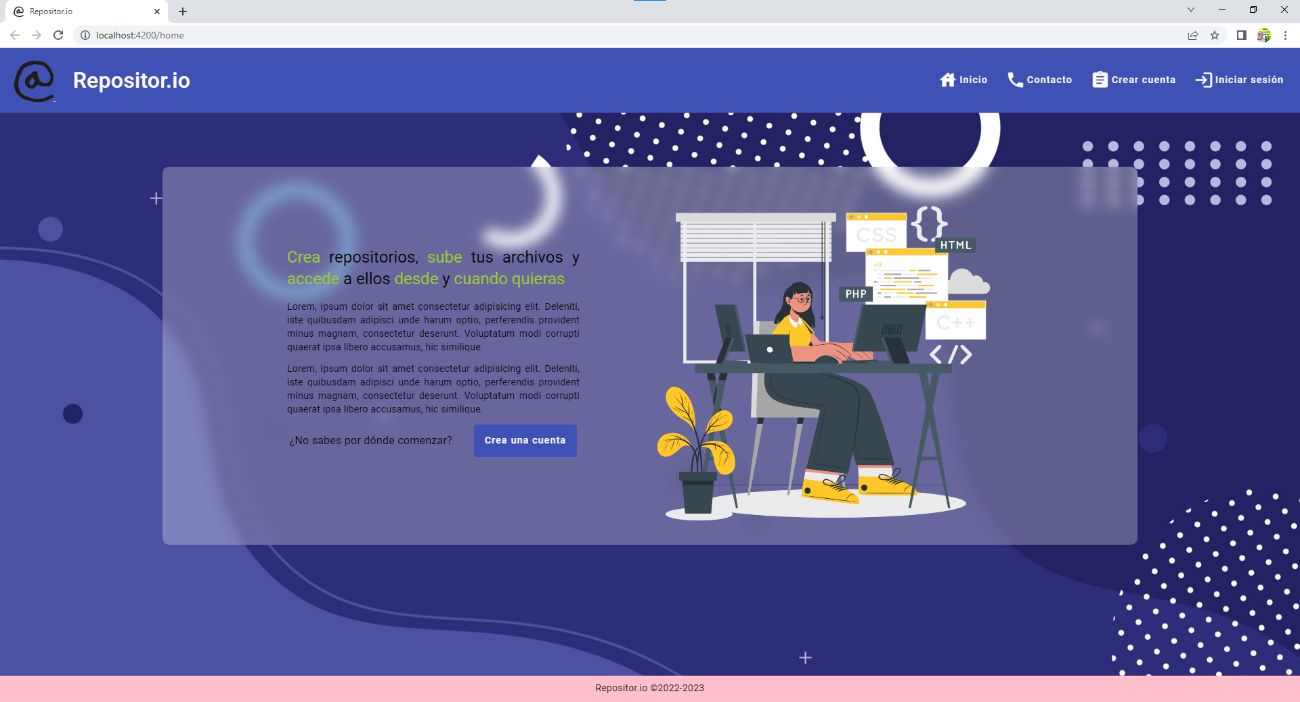


## VERSIÓN 1.0



# DISEÑO DE INTERFAZ

## TABLA DE IMÁGENES

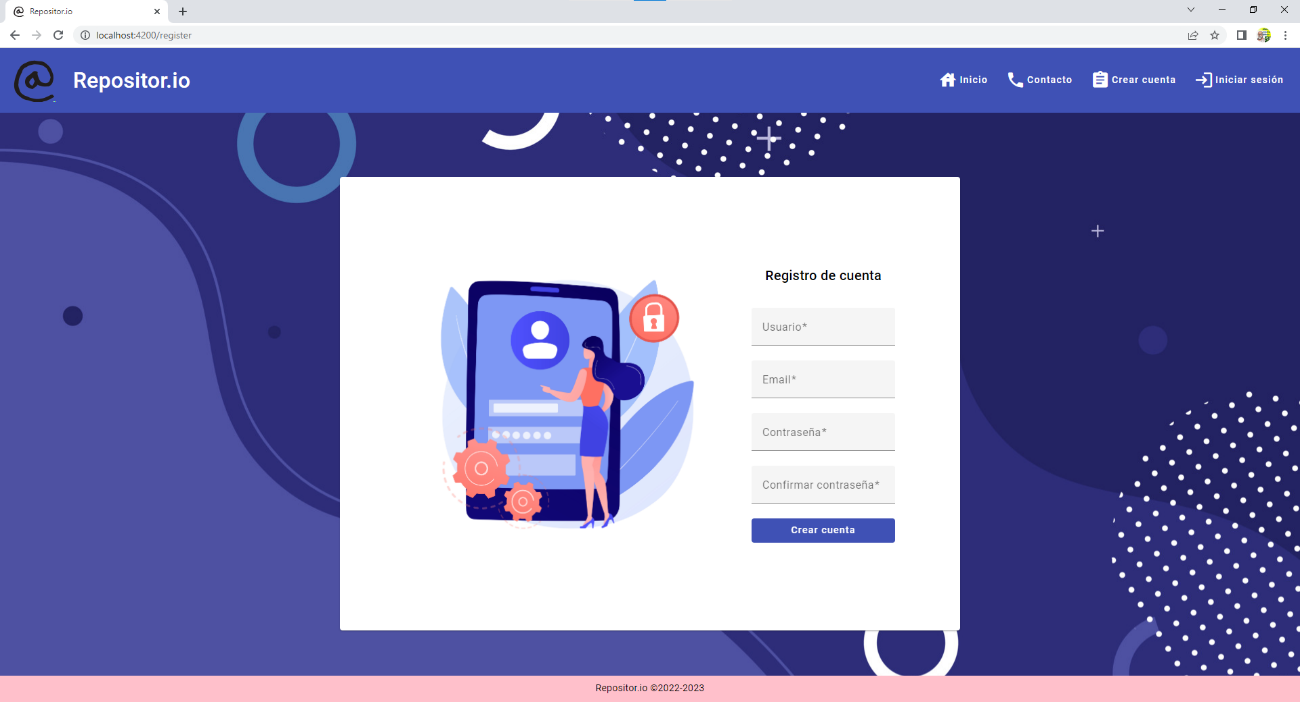
Figura 1: página de inicio de *Repositor.io*

Interfaz de usuario gráfica

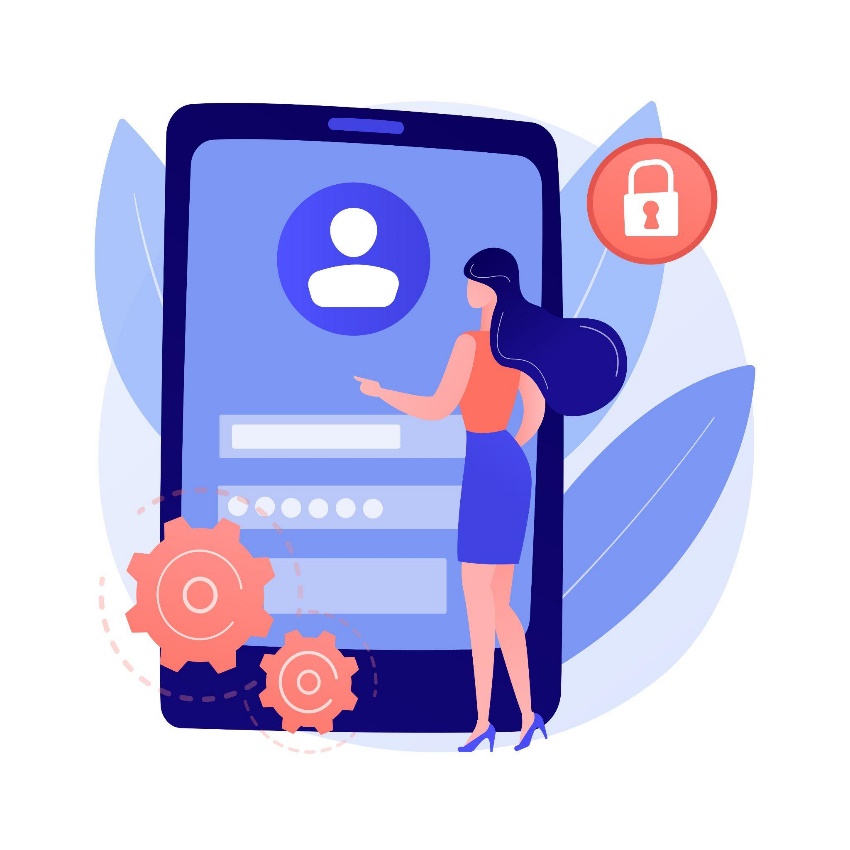
Descripción generada automáticamenteEsta captura de pantalla recoge la portada de la aplicación web, con la que se recibe a los usuarios. En ella, cualquier usuario puede crear una cuenta, acceder a una que haya establecido previamente y contactar con el equipo gestor de *Repositor.io*, para obtener ayuda o realizar una consulta de las dudas derivadas del uso.

Para que el diseño de la web resultase más atractivo a los usuarios, se ha incluido esta ilustración, obtenida del banco de imágenes con software de producción propia *Freepik*. Para hacer que el diseño de la web resultase ma´s atractivo y acogedor para el usuario, se ha incluido esta ilustració

Ilustración 1: creada con Freepik

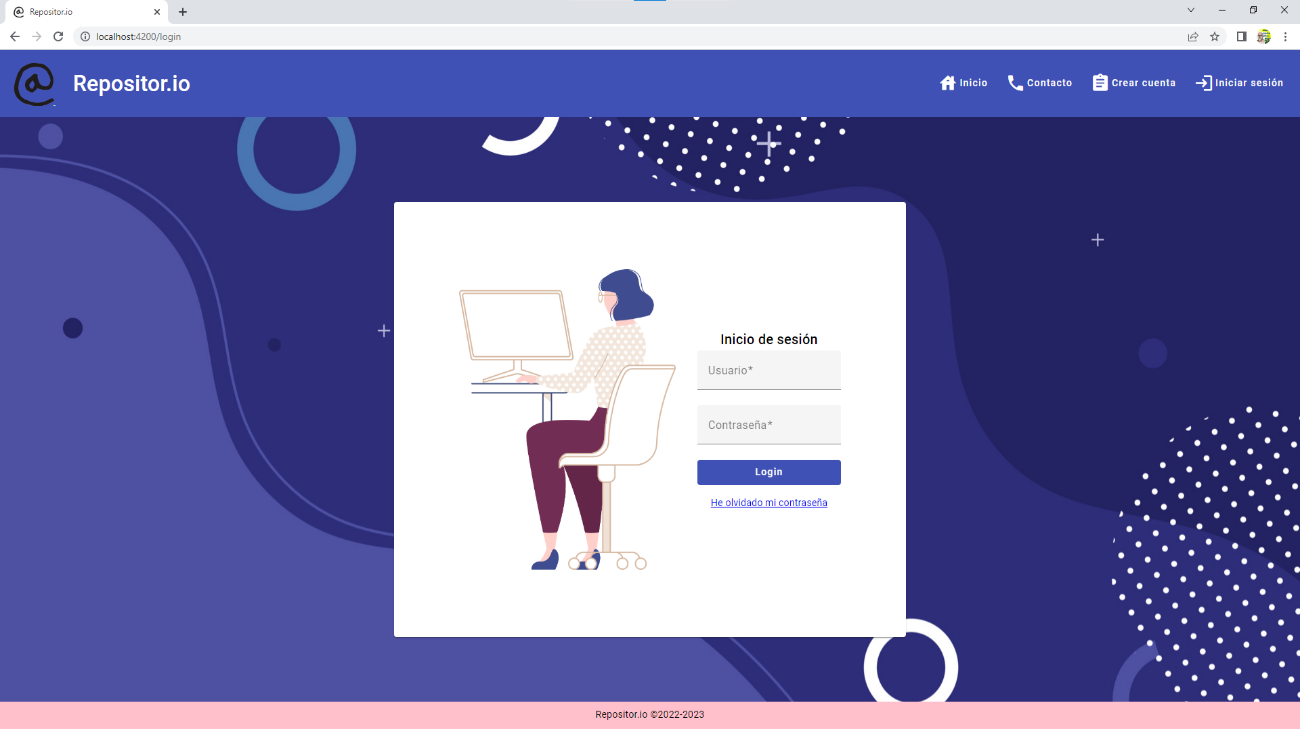
Figura 2: Registro de cuenta

Mediante cuatro sencillos pasos, el usuario puede crear su propia cuenta en la que crear y almacenar información a través de los repositorios.



De nuevo, atendiendo a los colores que se han empleado y con el objetivo de mantener cierta armonía estética, se han ido implementando diferentes ilustraciones que hacen referencia a las acciones que el usuario lleva a cabo en cada una de las diferentes opciones que permite la web. Al contar con un soporte visual que actúa de guía, se asegura también la accesibilidad.

Ilustración 2: creada con Freepik

Figura 3: Inicio de sesión de los usuarios

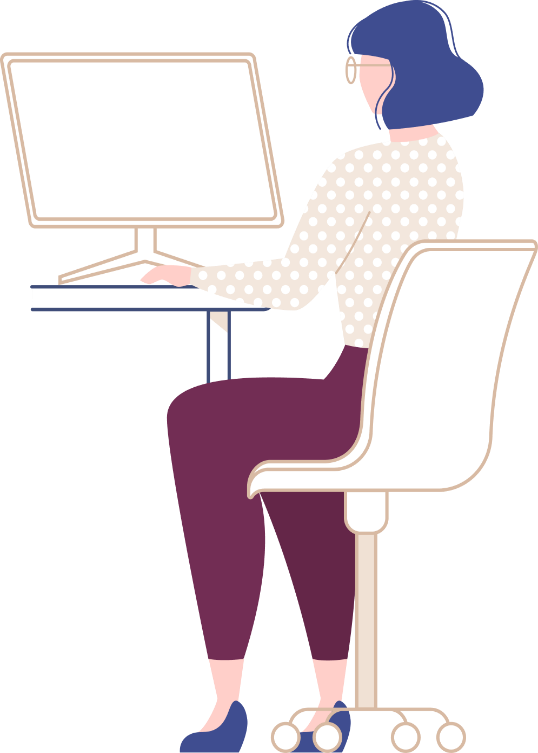
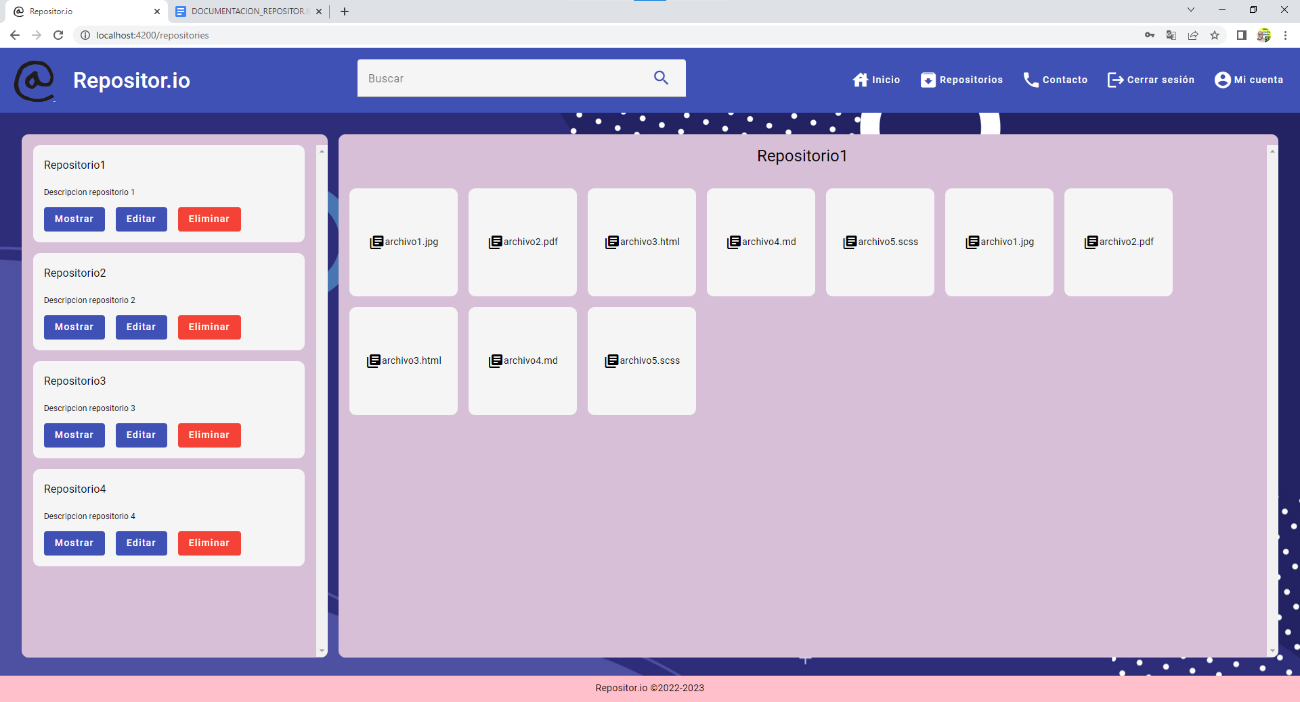
Una vez establecido el usuario, éste puede acceder directamente a su cuenta, teniendo incluso la opción de recuperar y modificar su contraseña, en caso de que se haya olvidado o se desee cambiar. Siguiendo la tónica de la web, se ha incluido una ilustración realizada con *Freepik*, que simboliza la acción del usuario.

Ilustración 3: creada con Freepik

Figura 4: Biblioteca de repositorios dentro de una cuenta de usuario

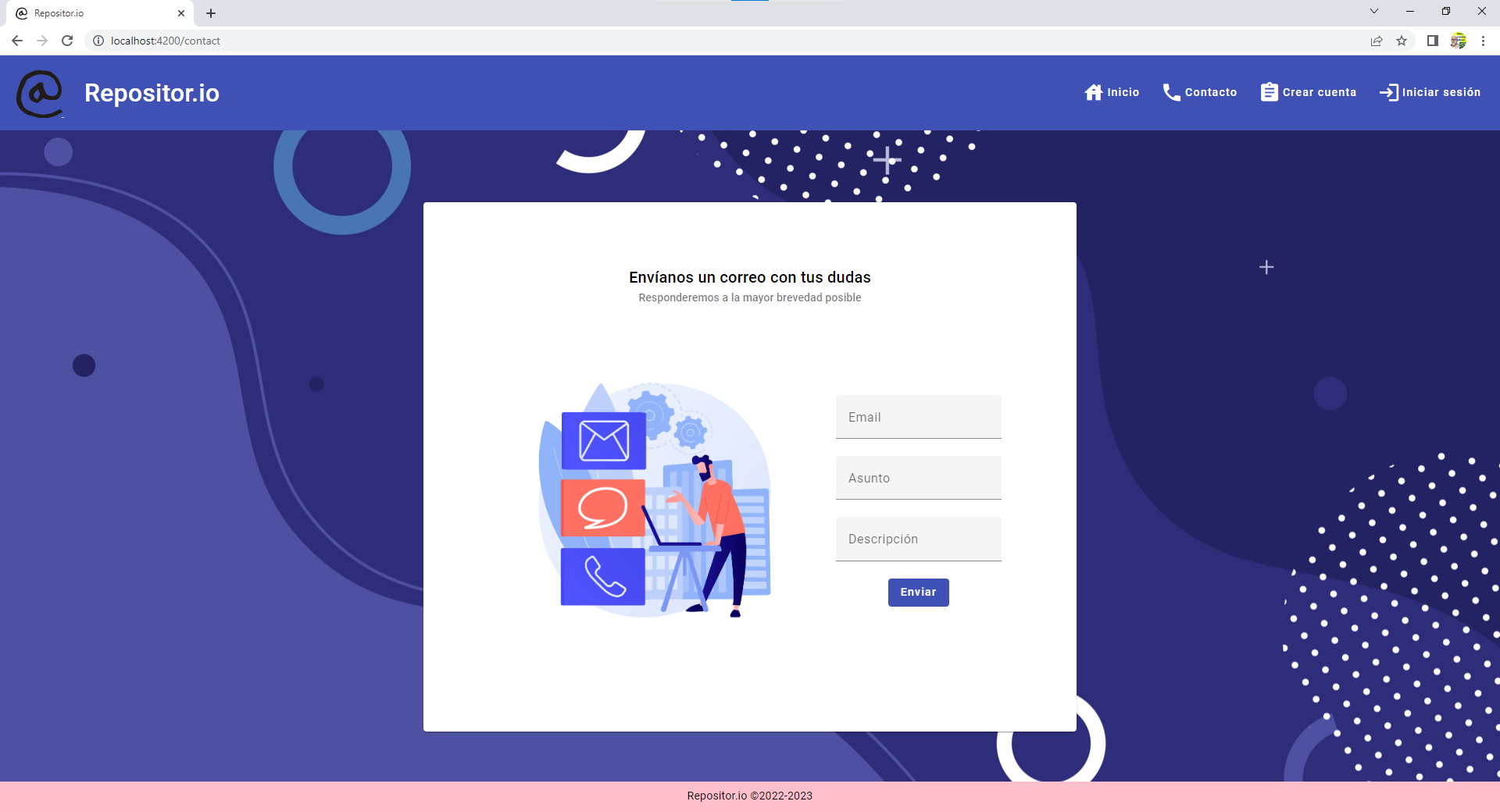


Este es el aspecto que tiene una biblioteca de repositorios dentro de la cuenta de cualquier usuario. A la izquierda de la pantalla, el usuario verá los diferentes repositorios que ha creado, los cuales podrá abrir, editar o eliminar, a su conveniencia. No existe un límite de repositorios que puedan crearse. Adicionalmente, se puede modificar el nombre del propio repositorio, permitiendo esto al usuario gestionar la información que desea almacenar en función de sus propios intereses y necesidades.

A la derecha de la pantalla, ocupando la mayor parte de la misma, el usuario tendrá acceso al contenido del repositorio seleccionado. Junto con el nombre identificativo, a cada archivo acompaña un icono que señala de qué tipo es, siendo este símbolo diferente para documentos y para imágenes, por ejemplo.

En esta imagen se aprecia también la barra de búsqueda, situada en el borde superior. Esta permite al usuario buscar a otros, con los que tenga interés en conectar.

Figura 5: Contacto y ayuda



El usuario podría contactar de manera directa con algún responsable del servicio técnico de *Repositor.io* para que éste resolviese sus dudas, indicando previamente una dirección de correo de contacto, el asunto de la comunicación, así como una descripción del problema.

Para esta sección, se ha empleado la siguiente imagen:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 4: creada con Freepik

Figuras 6 y 7: Fondo y otros elementos gráficos empleados:

Patrón de fondo

Descripción generada automáticamente



# 

Ilustración 5: personajes creados con Freepik

# GUÍA DE ESTILOS

En la creación de la interfaz del proyecto se han seguido unas pautas para lograr una estética común entre todos sus componentes. Este código visual está formado tanto por colores, fuentes como por el estilo de las imágenes usadas (ver [tabla de imágenes](#_35nkun2)).

**COLORES:**

* PRIMARIO: #3F51B5 ▇
* SECUNDARIO: #00C0CB ▇
* BLANCO: #F5F5F5 ▇
* GRIS: #D8BFD8 ▇
* ROJO: #F44336 ▇

**FUENTES TIPOGRÁFICAS:**

* Roboto light 400
* Roboto medium 600
* **Roboto strong 800**

**LOGOTIPO:**

**Logotipo

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente**Creado gracias a la Inteligencia Artificial *Dall-e 2*, su sencillez y familiaridad hacen que la asociación entre el símbolo y la aplicación web se produzca sin obstáculos en el imaginario del usuario. Además, su forma ligeramente orgánica aporta a la identidad de la web.

Ilustración 6: variación de color logo. Portada

# DISEÑO DE PRUEBAS

# WEBGRAFÍA

Angular (s.f). *Angular Docs*. <https://angular.io/docs>

Duffy, J. (12 de diciembre de 2004). Follow up: Should you invoke Close () and/or Dispose () on a Stream. *Joe Duffy’s Blog*. <https://joeduffyblog.com/2004/12/12/follow-up-should-you-invoke-close-andor-dispose-on-a-stream/>

MDN Web Docs (s.f). *References*. <https://developer.mozilla.org/es/docs/Web>

* *Guides.* [*https://developer.mozilla.org/es/docs/Learn*](https://developer.mozilla.org/es/docs/Learn)
* *Blog.* [*https://developer.mozilla.org/en-US/blog/*](https://developer.mozilla.org/en-US/blog/)

Tdykstra. (11 de abril de 2023). *APIs overview*. *Choose between controller-based APIs and minimal APIs*. Microsoft Learn. <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/apis?view=aspnetcore-7.0>