

Proyecto Entornos.

Jaime Sánchez Soteras.

Primera parte del proyecto:

DESCRIPCIÓN DETALLADA DE LA EMPRESA:

Nombre de la empresa: **Data Bloom Analytics (D B A)**

Componentes del grupo:

- **Rama de Dirección General**; encargada de gestionar, administrar y dar directrices al resto de componentes, responsables de las decisiones globales y rumbo de la empresa. Indispensable para el correcto funcionamiento de la empresa. Delega el resto de tareas en los diferentes departamentos y componentes de la empresa.
- **Rama de Desarrollo de Software**; grupo de expertos programadores que se encargan de crear las herramientas de análisis y visualización de datos, y las diferentes aplicaciones dadas en los servicios que ofrecemos. Se encargan de hacer mantenimiento de la vista que ve el usuario, y de la parte de procesamiento de trabajo, según una jerarquía de delegación de proyectos.
- **Rama de Ciencias Analíticas de Datos**; grupo de expertos analistas o científicos de datos que diseñan y contactan al grupo de Desarrollo de Software para crear algoritmos de interpretación de datos. Trata el apartado de detección de patrones, anomalías y predicciones simples.
- **Equipo de Infraestructura**; equipo de expertos con grandes dotes de adaptación que se encargan de asegurar que los servidores, bases de datos y conexiones a la nube siempre funcionen rápido, y seguro con disponibilidad plena.
- **Rama de Ventas y Marketing**; equipo de promoción del software, comunicándose con clientes y manejando redes sociales, página web, anuncios...
- **Atención al cliente y Soporte técnico**; amplio equipo de técnicos que ayudan a los usuarios cuando tienen problemas, dudas y buscan aprender a usar de mejor manera el producto.

- Equipo de mejora de Producto y Diseño; equipo encargado de que las plataformas sean sencillas, bonitas e intuitivas de usar para el usuario final, están en constante comunicación con la rama de Desarrollo de Software.
- Rama de finanzas y administración; equipo encargado del tema de cobros, pagos, contabilidad y cumplimiento legal.

Número de empleados por rama:

- | | |
|---|--------------------|
| - <u>Rama de Dirección General :</u> | 1 persona (C.E.O.) |
| - <u>Rama de Desarrollo de Software:</u> | 6-7 personas |
| - <u>Rama de Ciencias Analíticas de Datos:</u> | 3-5 personas |
| - <u>Equipo de Infraestructura :</u> | 2 personas |
| - <u>Rama de Ventas y Marketing:</u> | 3-4 personas |
| - <u>Atención al cliente y Soporte técnico:</u> | 4 personas |
| - <u>Equipo de mejora de Producto y Diseño:</u> | 2 personas |
| - <u>Rama de finanzas y administración:</u> | 3 personas. |

TOTAL: Entre 23-28 personas

Por cada 4 personas en un equipo, 1 de ellas es un coordinador.

Actividad de la empresa:

La empresa se dedica a desarrollar software de análisis y visualización de datos a pequeñas y medianas empresas.

Su actividad principal se basa en crear plataformas que permitan a los clientes conectar sus datos a estas, con la finalidad de que se procesen automáticamente, encontrando patrones o tendencias, mostrando los resultados y la información en paneles visuales fáciles de entender como gráficos o reportes dinámicos junto con posibilidad de recomendación de cambios. Estos análisis pueden ser compartidos en tiempo real entre equipos lo que les garantiza una mayor facilidad de comunicación y trabajo compartido.

Como ejemplo, una empresa de almacenes puede ver a tiempo real con un producto estándar contratado qué ventas están siendo más favorables y cuáles menos, viendo en un primer vistazo un gráfico. Además puede recibir alertas de Stock si baja demasiado o los plazos de caducidad de sus productos.

La empresa tiene preparados diferentes lotes de productos a la venta según las necesidades del cliente.

Una básica para pequeñas empresas, un paquete más completo para empresas medianas, y productos de mayor precio que son completamente customizables.

Sedes:

Su única sede actual se encuentra en **Baza**, pero va a extenderse en poco a comarcas cercanas.

Organigrama:

Data Bloom Analytics

Dirección
General

Desarrollo de
Software

Ventas y
Marketing

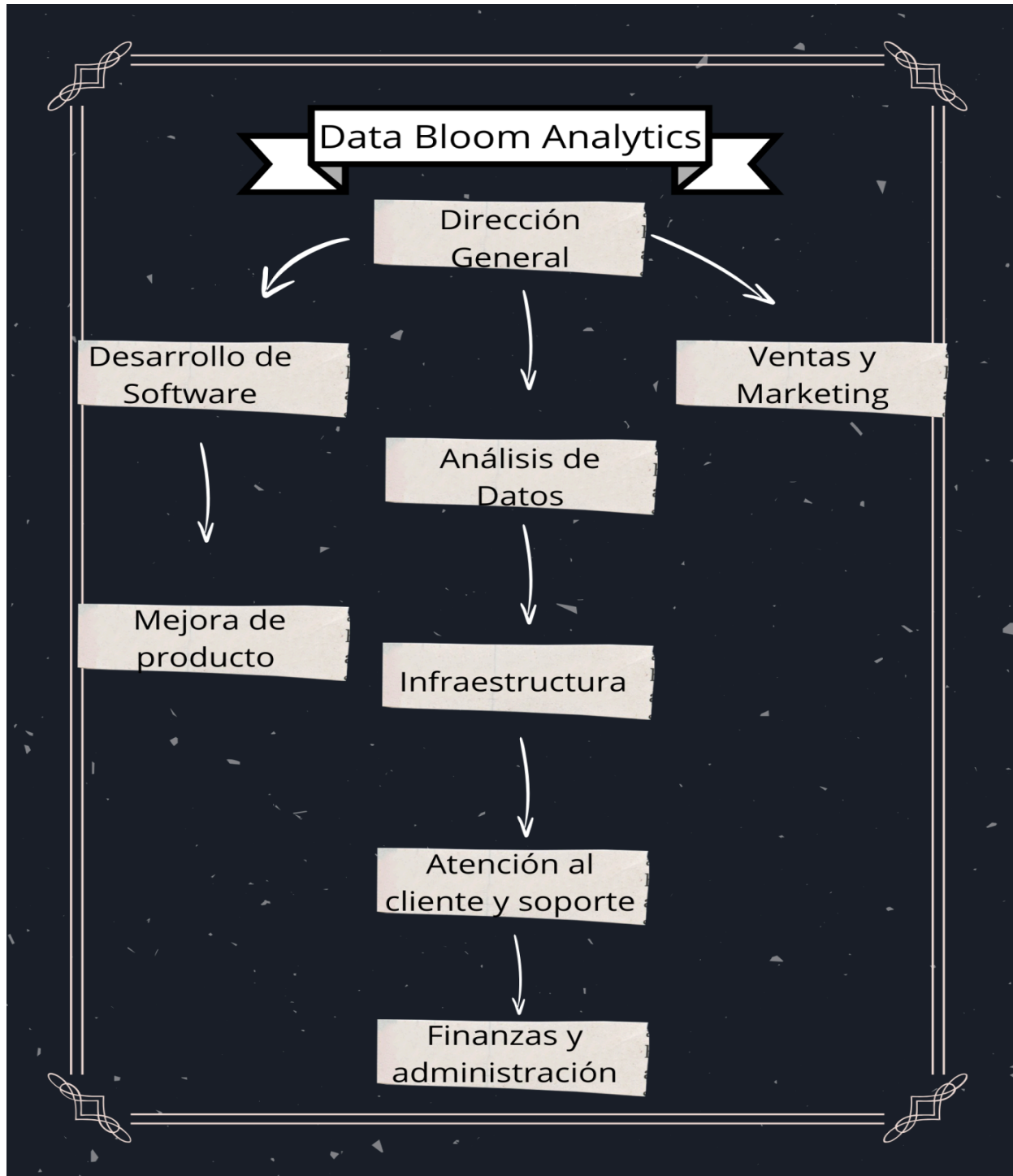
Análisis de
Datos

Mejora de
producto

Infraestructura

Atención al
cliente y soporte

Finanzas y
administración



Gestión de la información y nivel de informatización:

Data Bloom Analytics maneja un **nivel alto de informatización** debido a que su negocio es precisamente desarrollar y manejar sistemas de datos.

Toda la información empresarial sobre clientes, ventas, proyectos,..., está **almacenada en sistemas en la nube**.

Tienen bases de datos estructuradas para almacenar la información interna y de sus usuarios.

Su flujo de trabajo interno está también informatizado, utilizando herramientas SaaS para la comunicación, gestión de tareas y colaboración de los equipos.

El equipo de finanzas y administración también está informatizado, utilizando sistemas CRM y ERP.

Aplican la automatización de reportes, alertas inteligentes y sistemas de backup automáticos para proteger y gestionar sus datos.

Por supuesto cumplen con buenas prácticas de seguridad de la información con cifrado de datos, control de accesos y copias de seguridad regulares (semanales o mensuales)

DESCRIPCIÓN DE LOS RECURSOS DISPONIBLES EN LA EMPRESA

- **Rama de Dirección General :**

Hardware:

Ordenador portátil o de escritorio con acceso a internet y herramientas de comunicación.

Software:

Herramientas para gestión y planificación, sistemas de videoconferencia, correo electrónico y software para toma de decisiones.

- **Rama de Desarrollo de Software:**

Hardware:

Ordenadores potentes con procesadores rápidos y buena memoria para programar y ejecutar

pruebas de software.

Software:

Entornos de desarrollo (IDE), sistemas de control de versiones (como Git), herramientas para pruebas y gestión de proyectos ágiles.

- Rama de Ciencias Analíticas de Datos:

Hardware:

Ordenadores con capacidad para procesar grandes volúmenes de datos, con buena memoria y procesadores rápidos.

Software:

Programas y librerías para análisis de datos y creación de algoritmos, software estadístico y herramientas para comunicación con el equipo de desarrollo.

- Equipo de Infraestructura :

Hardware:

Servidores físicos y acceso a servidores en la nube, sistemas de almacenamiento, dispositivos de red (routers, switches).

Software:

Sistemas operativos de servidores, software de gestión de bases de datos, herramientas de monitorización, sistemas de seguridad y backup automáticos.

- Rama de Ventas y Marketing:

Hardware:

Ordenadores y dispositivos móviles para gestionar campañas y comunicación.

Software:

Plataformas para gestión de redes sociales, sistemas de email marketing, software CRM para gestionar clientes, página web corporativa.

- Atención al cliente y Soporte técnico:

Hardware:

Ordenadores y auriculares para atención telefónica y videollamadas.

Software:

Sistemas de ticketing para registrar y gestionar incidencias, plataformas de comunicación (chat, email), bases de datos con documentación y tutoriales.

- Equipo de mejora de Producto y Diseño:

Hardware:

Ordenadores con capacidades gráficas para diseño visual.

Software:

Programas de diseño gráfico, herramientas para creación de interfaces de usuario (UI/UX), software de prototipado y colaboración con desarrolladores.

- Rama de finanzas y administración:

Hardware:

Ordenadores para gestión administrativa y financiera.

Software:

Sistemas CRM y ERP para controlar cobros, pagos y contabilidad, programas de gestión documental y cumplimiento legal.

DESCRIPCIÓN DE LOS REQUISITOS PARA LA APLICACIÓN

La aplicación que se va a desarrollar en Data Bloom Analytics tiene como objetivo principal ofrecer a pequeñas y medianas empresas una plataforma que les permita conectar sus datos, analizarlos automáticamente y mostrar los resultados de forma visual e intuitiva.

Esto facilitará la toma de decisiones, el seguimiento de la actividad y la colaboración entre equipos en tiempo real.

Estos son los módulos que debe cubrir la aplicación junto a sus requisitos:

-En primer lugar, un **Módulo de Conexión y Gestión de Base de Datos.**

La aplicación debe permitir al usuario conectar diferentes fuentes de datos, como bases de datos internas, hojas de cálculo, o servicios en la nube.

Debe contar con un sistema seguro para gestionar las credenciales y permisos de acceso a estos datos.

Los datos importados deben ser validados para asegurar que sean correctos y estén completos antes de iniciar su procesamiento.

-Además, un **Módulo de Procesamiento y Análisis de Datos.**

La plataforma debe procesar automáticamente los datos recibidos para identificar patrones, tendencias y anomalías.

Debe permitir la ejecución de algoritmos personalizados para diferentes tipos de análisis, según las necesidades del cliente.

Este módulo debe ser capaz de manejar grandes volúmenes de datos y realizar cálculos en tiempo razonable.

-Además un **Módulo de Visualización.**

La aplicación debe mostrar los resultados del análisis en paneles visuales que incluyan gráficos interactivos, tablas dinámicas y reportes personalizados.

Los usuarios deben poder configurar qué datos y métricas quieren ver y cómo desean que se muestren.

Las visualizaciones deben actualizarse en tiempo real o con la frecuencia que el cliente configure.

-Además un **Módulo de Alertas y Notificaciones.**

El sistema debe generar alertas automáticas cuando se detecten situaciones importantes, como disminución de stock o anomalías en las ventas.

Estas alertas deben poder enviarse por diferentes canales: dentro de la aplicación, por correo electrónico o mediante notificaciones móviles.

Los usuarios deben poder configurar las reglas que disparan estas alertas según sus necesidades.

-Además un **Módulo de Colaboración.**

La plataforma debe permitir que varios usuarios puedan acceder a los mismos datos y análisis, facilitando el trabajo en equipo.

Debe incluir funciones para compartir paneles y reportes con otros usuarios, con diferentes niveles de permisos (solo lectura, edición, etc.).

Se debe contar con un sistema de comentarios o notas para que los usuarios puedan comunicarse directamente sobre los análisis.

-Además un **Módulo de Administración y Seguridad.**

Debe existir un sistema para gestionar los usuarios y sus roles, controlando quién puede acceder a qué información y funcionalidades.

La aplicación debe incluir medidas de seguridad, como cifrado de datos y control de accesos.

Se deben implementar copias de seguridad automáticas para evitar pérdidas de información.

-Por último un Módulo de Personalización y Configuración.

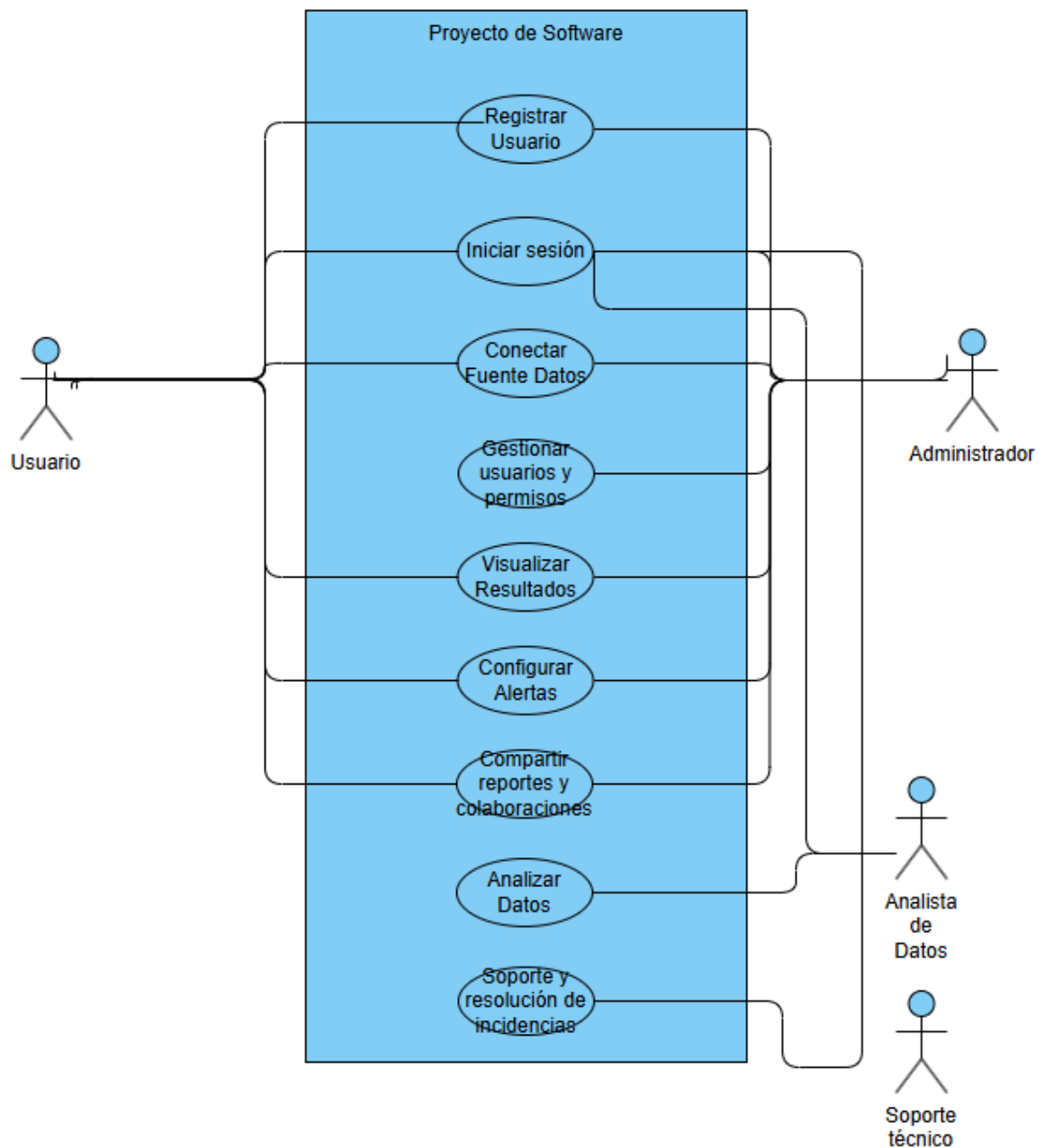
Los clientes deben poder elegir entre diferentes planes o productos según sus necesidades, desde opciones básicas hasta soluciones personalizadas.

La aplicación debe permitir configurar aspectos como la frecuencia de actualización de datos, idiomas y temas visuales.

Se debe facilitar la integración con otros sistemas que el cliente utilice.

Segunda parte del proyecto:

Casos de uso del Proyecto de Software.



Descripción de los Actores:

Administrador: Persona encargada de administrar la plataforma, gestionar usuarios, asignar roles y mantener la seguridad y configuraciones generales del sistema.

Usuario Cliente: Usuario final que utiliza la plataforma para cargar sus datos, analizar y visualizar información, configurar alertas y colaborar con otros usuarios.

Analista de Datos: Profesional que diseña y adapta los algoritmos de análisis para mejorar los resultados y la detección de patrones en los datos.

Soporte Técnico: Equipo encargado de atender las consultas, problemas técnicos y dar soporte para el correcto uso de la aplicación.

Descripción de los Casos de Uso:

Primer Caso:

Registrar Usuario

Lo realiza el Usuario.

Permite que un nuevo usuario cree una cuenta para acceder a la plataforma.

Secuencia de pasos:

1-El usuario accede al formulario de registro.

2-Introduce sus datos personales y de contacto.

3-El sistema valida que los datos sean correctos y que el usuario no esté registrado previamente.

4-El sistema crea la cuenta y envía un correo de confirmación.

5-El usuario confirma la cuenta a través del enlace recibido.

Posibles excepciones:

- Si el correo ya está registrado, se muestra un mensaje de error y se solicita otro correo.
- Si los datos están incompletos o no cumplen el formato, el sistema pide corregirlos.

Segundo Caso:

Iniciar Sesión

Lo realizan todos los actores.

Permite la entrada a la plataforma en base a unas credenciales.

Secuencia de pasos:

- 1-El usuario introduce su correo y contraseña.
- 2-El sistema verifica las credenciales.
- 3-Si son correctas, se abre la sesión y muestra la página principal.
- 4-Si el usuario es administrador, se muestran opciones avanzadas.

Posibles excepciones:

- Si las credenciales son incorrectas, se muestra un mensaje de error.
 - Si el usuario no ha confirmado su correo, se solicita confirmar antes de iniciar sesión.
-

Tercer Caso:

Conectar Fuente de Datos

Lo realiza el Usuario.

Permite al usuario conectar sus bases de datos o archivos para que la aplicación pueda procesar la información.

Secuencia de pasos:

- 1-El usuario accede al módulo de conexión de datos.
- 2-Selecciona el tipo de fuente de datos (base de datos, archivo Excel, nube, etc.).
- 3-Introduce las credenciales o carga el archivo correspondiente.
- 4-El sistema valida el acceso y los datos.
- 5-Si todo es correcto, la fuente queda conectada para su análisis.

Posibles excepciones:

- Si las credenciales son incorrectas, el sistema avisa y pide corregirlas.
 - Si el formato del archivo no es válido, se rechaza la conexión.
-

Cuarto Caso:**Conectar Fuente de Datos**

Lo realiza el Usuario.

Permite al usuario ver los resultados del análisis en forma de gráficos, tablas y reportes.

Secuencia de pasos:

- 1-El usuario selecciona el panel de visualización.
- 2-Escoge las métricas o datos que desea mostrar.

3-La aplicación genera gráficos y reportes en tiempo real.

4-El usuario puede personalizar la vista (tipo de gráfico, filtros, etc.).

Posibles excepciones:

- Si no hay datos disponibles o la conexión falla, se muestra un mensaje de error.
- Si el usuario intenta acceder a datos para los que no tiene permiso, la aplicación bloquea el acceso.

Quinto Caso:

Gestionar usuarios y permisos.

Lo realiza el Administrador.

Permite al administrador tener un control de cuentas de usuario existentes, dejándole editar campos o borrar registros en caso necesario.

Secuencia de pasos:

1-El administrador inicia sesión con su cuenta

2-Abre el menú de usuarios y permisos

3-Elige que usuario quiere modificar

4-Cambia los datos o borra un usuario

5-Si todo es correcto, la operación se completa.

Posibles excepciones:

- Si los campos rellenos no son correctos, el sistema avisa y pide que los cambie.
 - Si se le intenta asignar permisos que no debería tener a un usuario corriente se pedirá una confirmación extra.
-

Sexto Caso:

Configurar Alertas

Lo realiza tanto el Usuario como el Administrador.

Permite al usuario programar alertas personalizadas para su software, como por ejemplo, si quiere que se le avise cuando se hayan cargado nuevos datos, o cuánto ha tardado en ejecutar la acción. El administrador también puede configurar nuevas alertas para la aplicación base, como cuánto queda para el cobro, las actualizaciones...

Secuencia de pasos:

- 1-El usuario o administrador abren la plataforma.
- 2-Acceden al menú “Alertas”.
- 3-Eligen si quieren crear o editar una alerta, en caso de ser Administrador las opciones son distintas.
- 4-El sistema valida que la alerta sea válida
- 5-Si todo es correcto, la alerta queda configurada correctamente.

Posibles excepciones:

· Si los valores para crear la alerta no son correctos, se pedirá revisar y no funcionará la alerta.

Séptimo Caso:

Compartir reportes y colaboraciones

Lo realizan tanto el Usuario como el Administrador.

Permite al usuario y al administrador comunicarse de manera más sencilla, para solucionar errores, o pedir cambios necesarios.

Secuencia de pasos:

1-El usuario o Administrador accede al módulo de comunicación.

2-Selecciona el tipo de mensaje que quiere elegir.

3-Introduce los datos necesarios.

4-El sistema valida que el mensaje sea claro, y se estudia su petición.

5-Si todo es correcto, la administración se pondrá en contacto con el usuario para verificar si los cambios son posibles.

Octavo Caso:

Analizar Datos

Lo realiza el Analista de Datos.

Permite al Analista de Datos ejecutar su función, con un módulo al cual sólo él tiene acceso.

Secuencia de pasos:

1-El Analista accede al módulo de conexión de datos.

2-Recibe un informe de datos semanal, mensual o trimestral.

3-Estudia los datos.

4-Envía un reporte al resto del equipo.

Posibles excepciones:

- Si no hay datos visibles por algún error, el analista dará un aviso inmediato.

Noveno Caso:

Soporte y resolución de incidencias

Lo realiza el Soporte técnico.

Permite al Soporte Técnico leer los reportes de los usuarios, para administrar incidencias, corregir errores, y dar apoyo y atención inmediata a los usuarios que lo necesiten.

Secuencia de pasos:

1-El Soporte técnico accede a los archivos de reporte.

2-Estudian los casos.

3-Se ponen en contacto con el usuario.

4-Prestan servicio, y en caso de ser necesario, contactan con el equipo de administración.

Repositorio de Github para el Prototipo:

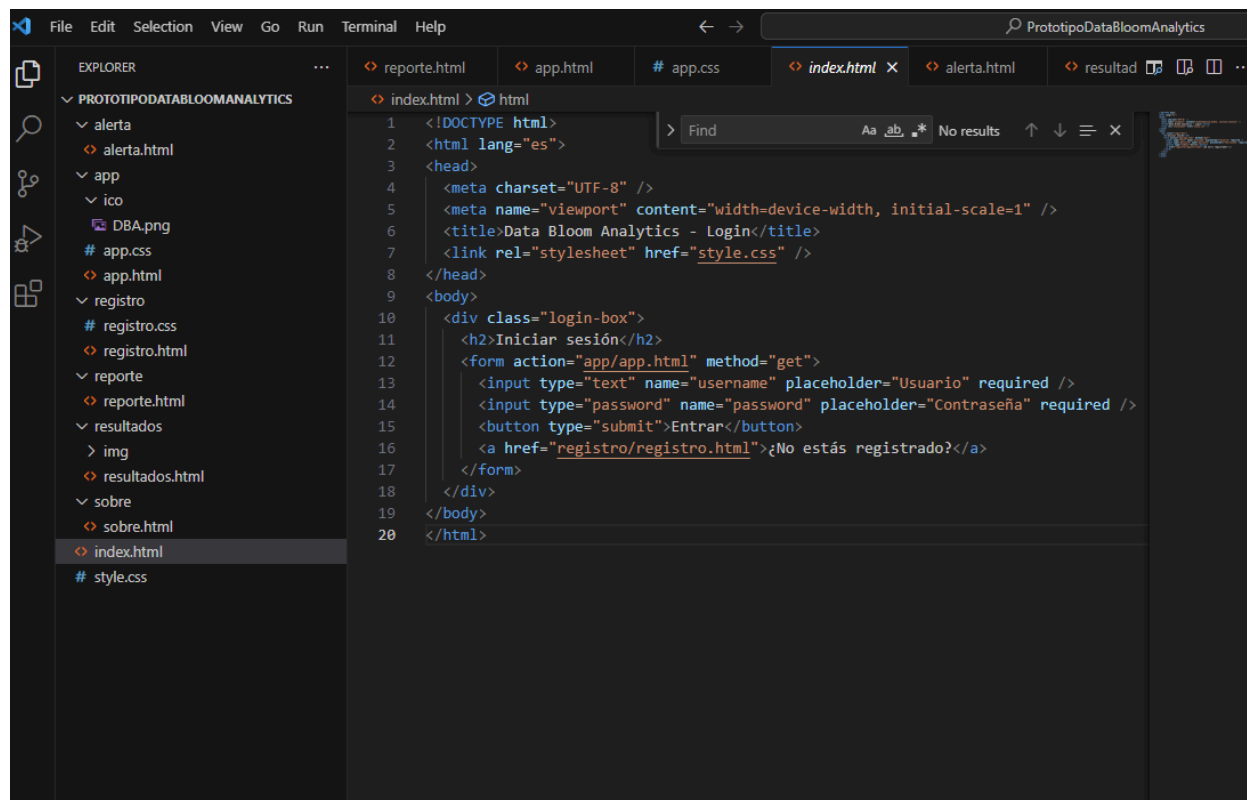
Enlace:

<https://github.com/JaimeSanchez75/DataBloomAnalytics.git>

Página del Github:

<https://jaimesanchez75.github.io/DataBloomAnalytics/>

Paso 1: Crear el proyecto e ir trabajando en el.

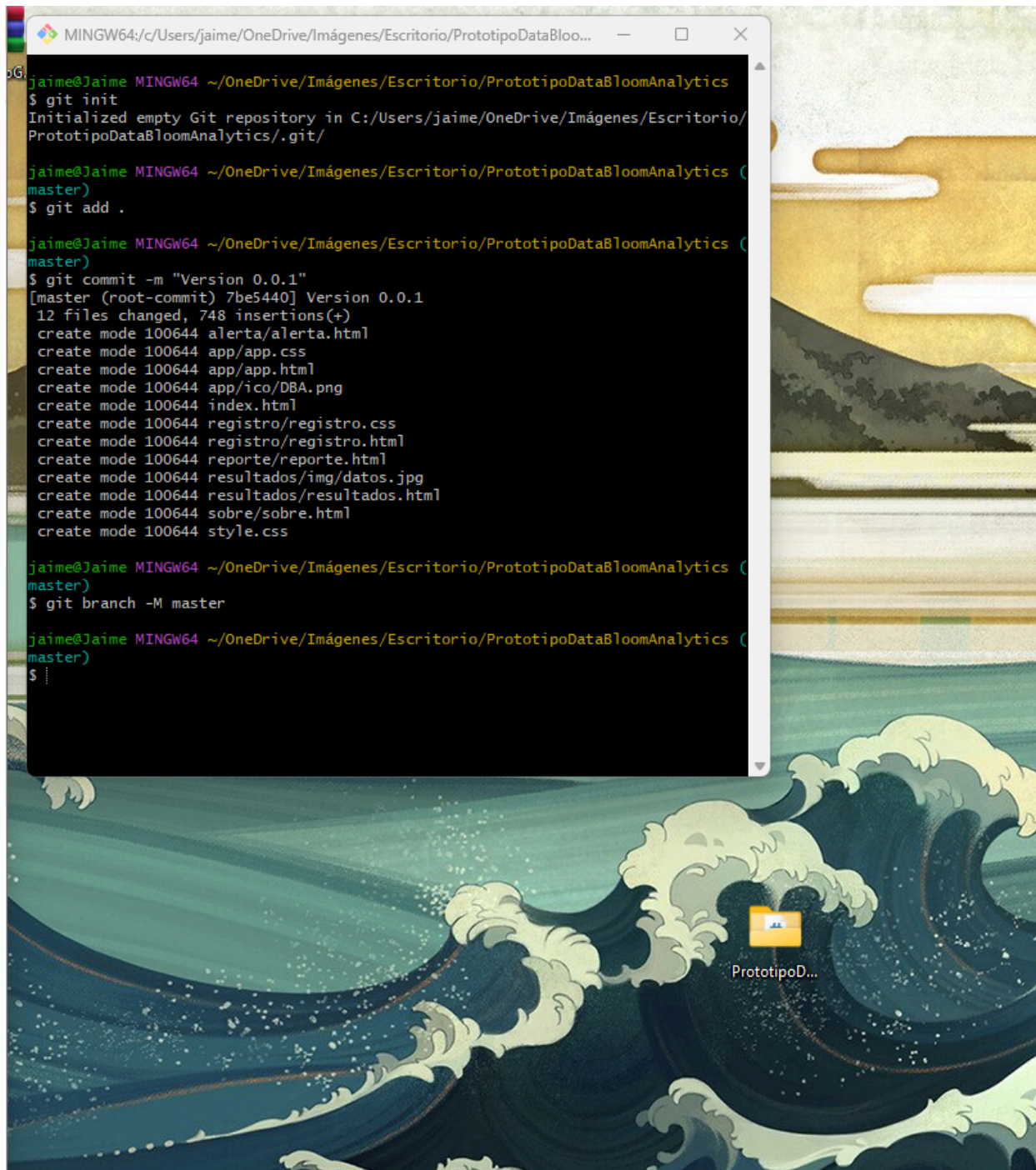


The screenshot shows a Visual Studio Code editor with a project named 'PROTOTIPODATABLOOMANALYTICS'. The Explorer sidebar on the left shows the project structure, including folders like 'alerta', 'app', 'registro', 'reporte', 'resultados', and 'sobre', and files like 'index.html' and 'style.css'. The main editor area displays the content of 'index.html', which is an HTML document for a login page. The code includes a DOCTYPE declaration, HTML lang attribute, head section with meta tags for charset and viewport, and a link to 'style.css'. The body contains a 'login-box' div with a heading 'Iniciar sesión', a form with username and password inputs, a submit button, and a link to the registration page.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1" />
6   <title>Data Bloom Analytics - Login</title>
7   <link rel="stylesheet" href="style.css" />
8 </head>
9 <body>
10  <div class="login-box">
11    <h2>Iniciar sesión</h2>
12    <form action="app/app.html" method="get">
13      <input type="text" name="username" placeholder="Usuario" required />
14      <input type="password" name="password" placeholder="Contraseña" required />
15      <button type="submit">Entrar</button>
16      <a href="registro/registro.html">¿No estás registrado?</a>
17    </form>
18  </div>
19 </body>
20 </html>
```

terminado.

Paso 2: añadir la versión Prototipo a Github:



```
MINGW64:/c:/Users/jaime/OneDrive/Imágenes/Escritorio/PrototipoDataBloo...
jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git remote add jaime https://github.com/JaimeSanchez75/DataBloomAnalytics.git

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git push -u origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git push -u jaime master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (22/22), 232.94 KiB | 25.88 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/JaimeSanchez75/DataBloomAnalytics.git
 * [new branch]      master -> master
branch 'master' set up to track 'jaime/master'.
```

```
MINGW64:/c:/Users/jaime/OneDrive/Imágenes/Escritorio/PrototipoDataBloo...
create mode 100644 registro/registro.html
create mode 100644 reporte/reporte.html
create mode 100644 resultados/img/datos.jpg
create mode 100644 resultados/resultados.html
create mode 100644 sobre/sobre.html
create mode 100644 style.css

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git push -u jaime master --force
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (22/22), 232.95 KiB | 23.29 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/JaimeSanchez75/DataBloomAnalytics.git
 r f89431f...a0b3363 master -> master (forced update)
branch 'master' set up to track 'jaime/master'.
```

github.com/JaimeSanchez75/DataBloomAnalytics

JaimeSanchez75 / DataBloomAnalytics

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

DataBloomAnalytics Public

Pin Watch 0 Fork 0 Star 0

master

Go to file + Code

About

Projecto prototipo de un Software de Base de Datos

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published











Publish your first package

	Version 0.0.1	a0b3363 · now
alerta	Version 0.0.1	now
app	Version 0.0.1	now
registro	Version 0.0.1	now
reporte	Version 0.0.1	now
resultados	Version 0.0.1	now
sobre	Version 0.0.1	now
index.html	Version 0.0.1	now
style.css	Version 0.0.1	now

 alerta	Version 0.0.2 - Mejora visual	1 hour ago
 app	Version 0.0.3 - Mejora Visual General	30 minutes ago
 datos	Version 0.0.3 - Mejora Visual General	30 minutes ago
 registro	Version 0.0.1	12 hours ago
 reporte	Version 0.0.2 - Mejora visual	1 hour ago
 resultados	Version 0.0.3 - Mejora Visual General	30 minutes ago
 sobre	Version 0.0.2 - Mejora visual	1 hour ago
 index.html	Version 0.0.1	12 hours ago
 style.css	Version 0.0.1	12 hours ago

1 Manual del Usuario:
2 Última actualización: 29/05/2025.
3
4 Buenas! Esta es una guía de uso de la aplicación Data Bloom Analytics en su Plataforma Web de Software de gestión y administración de bases de datos.
5
6 El programa empieza con la ejecución de el archivo "index.html".
7
8 La ejecución de éste abrirá una página de Inicio de Sesión. En caso de disponer de datos (Simulado, realmente no hay forma de iniciar sesión realmente) introdúzcalos y pruebe a iniciar sesión.
9 Si los datos son correctos, le llevará a la página principal de la plataforma.
10
11 En caso de no disponer de datos de inicio de sesión, le recomendamos darle al apartado de registro (También simulado, no hay forma de registrarse realmente).
12
13 Al acceder a la página principal verá un apartado de notificaciones sin ver en la bandeja de entrada, junto con un apartado de parches de actualización.
14
15 En el lateral izquierdo se mostrará un menú de opciones para desplazarse por la aplicación, como un apartado con información sobre Data Bloom Analytics en el nombre de la empresa, un apartado para configurar notificaciones,
16 un apartado para generar un reporte y contactar con el soporte, un apartado para ver un informe de datos actualizados, y por último un apartado para volcar o actualizar datos de la base de datos de cada usuario.
17
18 En el apartado de Avisos y Notificaciones podrá encontrar un formulario con el tipo de notificación que quiere crear, cada cuanto aparece, y el texto de la notificación, totalmente personalizable.
19
20 En el apartado de Reportes e Incidencias podrá encontrar un formulario con el tipo de reporte que necesita, un correo de contacto para que puedan enviarle también una notificación al correo, y por último el mensaje para enviar al soporte.
21 Aparecen las instrucciones un poco más abajo del formulario.
22
23 En el apartado de Informe de Resultados podrá encontrar un gráfico (Simulado) representando la Base de Datos, un resumen de datos, e información útil sobre ésta.
24
25 Por último, en el apartado de Conexión a la Fuente de Datos, encontrará un apartado para volcar ficheros datasource individuales, volcar conjuntos de datos, descargar copias de seguridad entre algunas de las funciones.

 JaimeSanchez75 Create README.md

Name	Last commit message
 alerta	Version 0.0.2 - Mejora visual
 app	Version 0.0.3 - Mejora Visual General
 datos	Version 0.0.3 - Mejora Visual General
 registro	Version 0.0.1
 reporte	Version 0.0.2 - Mejora visual
 resultados	Version 0.0.3 - Mejora Visual General
 sobre	Version 0.0.2 - Mejora visual
 README.md	Create README.md
 index.html	Version 0.0.1
 style.css	Version 0.0.1

MINGW64:/c/Users/Javi/Desktop/JaimeSánchezSoteras/PaginaDBA

```
DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.84 KiB | 209.00 KiB/s, done.
From https://github.com/JaimeSanchez75/DataBloomAnalytics
   89031cd..fbfefe8 master    -> jaimee/master
Updating 89031cd..fbfefe8
Fast-forward
 README.md | 25 ++++++
 1 file changed, 25 insertions(+)
 create mode 100644 README.md

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (master)
$ git checkout -b other-data
Switched to a new branch 'other-data'

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (other-data)
$ |
```

MINGW64:/c/Users/Javi/Desktop/JaimeSánchezSoteras/PaginaDBA

```
to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <>) not allowed

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (other-data)
$ git config user.email "jaime.sanchez.soteras@gmail.com"

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (other-data)
$ git config user.name "JaimeSanchez75"

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (other-data)
$ git commit -m"Version 0.1.0 - PDF Enviados anteriormente"
[other-data 7a16d77] Version 0.1.0 - PDF Enviados anteriormente
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 other-data/Casos de uso del Proyecto de Software.pdf
 create mode 100644 other-data/Proyecto de Software_JSS.pdf

DESKTOP-4RSI81G+Javi@DESKTOP-4RSI81G MINGW64 ~/Desktop/JaimeSánchezSoteras/PaginaDBA (other-data)
$
```



JaimeSanchez75 commented 1 minute ago

Owner ...

No description provided.



Version 0.1.0 - PDF Enviados anteriormente

7a16d77



JaimeSanchez75 merged commit **ef759b0** into **master** 1 minute ago

Revert



This branch has not been deployed

No deployments

Pull request successfully merged and closed

You're all set — the **other-data** branch can be safely deleted.

Delete branch



Add a comment

Write

Preview

H

B

I

≡

<>

🔗

≡

≡

≡

📎

...

Add your comment here...

Markdown is supported

Paste, drop, or click to add files

Comment



JaimeSanchez75 Version 0.1.0 - PDF Enviados anteriormente

7a16d77 · 3 minutes ago



Name	Last commit message	Last commit date
..		
Casos de uso del Proyecto de Software.pdf	Version 0.1.0 - PDF Enviados anteriormente	3 minutes ago
Proyecto de Software_JSS.pdf	Version 0.1.0 - PDF Enviados anteriormente	3 minutes ago

Código de un Módulo del programa

Módulo elegido:

Creación de Reportes e Incidencias.

Parte 1:

```
public class ReporteSoporte {  
  
    // Atributos  
    private String correo;  
    private String tipoReporte;  
    private String mensaje;  
    private String estado; // Por ejemplo: "Pendiente", "En proceso", "Resuelto"  
    private String fechaEnvio;  
  
    // Constructor  
    public ReporteSoporte(String correo, String tipoReporte, String mensaje, String fechaEnvio) {  
        this.correo = correo;  
        this.tipoReporte = tipoReporte;  
        this.mensaje = mensaje;  
        this.estado = "Pendiente";  
        this.fechaEnvio = fechaEnvio;  
    }  
  
    // Método implementado  
    public String generarConfirmacion(String nombreUsuario) {  
        return "Hola " + nombreUsuario + ", hemos recibido tu reporte sobre \""  
            + tipoReporte + "\". Nuestro equipo de soporte te contactará al correo "  
            + correo + " en un plazo de 48 horas.";  
    }  
  
    // Getters y setters (no es obligatorio implementarlos todos para este ejercicio)  
    public String getCorreo() {  
        return correo;  
    }  
  
    public void setCorreo(String correo) {  
        this.correo = correo;  
    }  
  
    public String getTipoReporte() {  
        return tipoReporte;  
    }  
  
    public void setTipoReporte(String tipoReporte) {  
        this.tipoReporte = tipoReporte;  
    }  
  
    public String getMensaje() {  
        return mensaje;  
    }  
  
    public void setMensaje(String mensaje) {  
        this.mensaje = mensaje;  
    }  
}
```

```

    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getFechaEnvio() {
        return fechaEnvio;
    }

    public void setFechaEnvio(String fechaEnvio) {
        this.fechaEnvio = fechaEnvio;
    }
}

```

Cuando el usuario accede a la sección de "Compartir reportes", rellena un formulario con su correo, el tipo de problema y un mensaje descriptivo. Al pulsar el botón de "Enviar", los datos se envían al servidor, donde se procesa la información.

En la parte del backend, se crea una instancia de la clase **ReporteSoporte** utilizando los datos del formulario. Esta clase contiene atributos como el correo electrónico, el tipo de reporte, el mensaje del usuario, la fecha de envío y el estado del reporte.

Una vez creado el objeto, se puede invocar el método **generarConfirmacion**, que recibe el nombre del usuario y devuelve un mensaje confirmando la recepción del reporte. Este mensaje puede mostrarse en la interfaz o enviarse por correo como acuse de recibo.

Después de eso, el reporte puede almacenarse en el sistema para que el equipo de soporte lo revise y gestione según su estado. Aunque esta parte no está implementada, la clase está preparada para integrarse con procesos de almacenamiento, seguimiento y respuesta.

Parte 2:

```

import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class ReporteSoporteTest {

    @Test
    public void testGenerarConfirmacion() {
        // Datos de entrada simulados
        ReporteSoporte reporte = new ReporteSoporte(
            "usuario@ejemplo.com",
            "Problemas inicio de sesión y permisos",
            "No puedo iniciar sesión desde ayer.",
            "2025-05-29"
        );

        // Ejecución del método con parámetro
        String resultado = reporte.generarConfirmacion("Carlos");

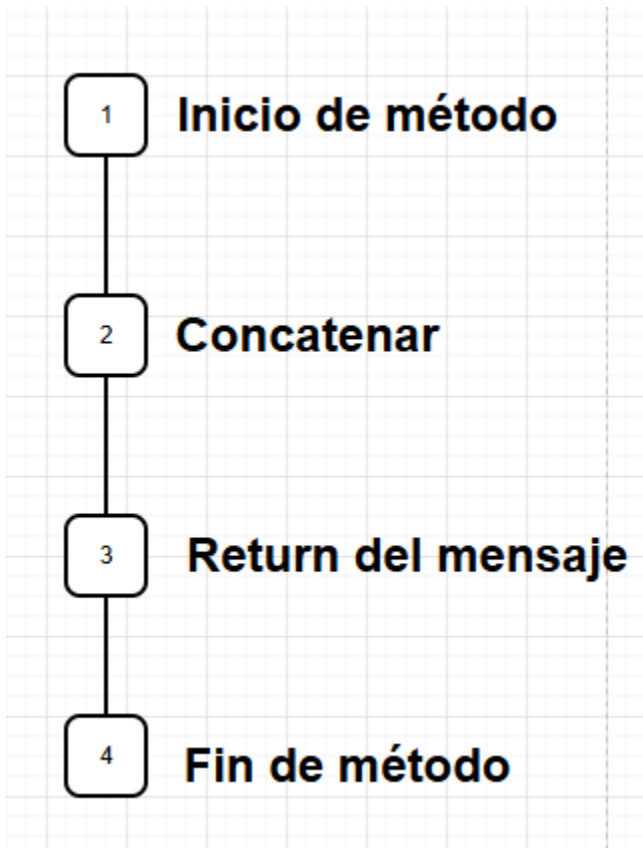
        // Resultado esperado
        String esperado = "Hola Carlos, hemos recibido tu reporte sobre \"Problemas inicio de sesión y permisos\". Nuestro equipo de soporte te contactará al correo usuario@ejemplo.com en un plazo de 48 horas.";

        // Verificación
        assertEquals(esperado, resultado);
    }
}

```

Prueba Junit caja negra. El resultado es el esperado, al llamarlo crea un reporte con los datos que se introduzcan en el formulario, y podemos verificar que la salida es exactamente lo esperado.

Prueba Junit caja blanca. El método al ser muy sencillo no tiene ramas secundarias con bucles o condiciones, por lo tanto tiene un esquema muy básico por lo que sería:



```
MINGW64/c/Users/jaime/OneDrive/Imágenes/Escritorio/PrototipoDataBloo...
jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git pull
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 18 (delta 7), reused 13 (delta 5), pack-reused 0 (from 0)
Unpacking objects: 100% (18/18), 958.45 KiB | 2.90 MiB/s, done.
From https://github.com/JaimeSanchez75/DataBloomAnalytics
   cb2e987..ef759b0  master       -> jaime/master
   * [new branch]   other-data -> jaime/other-data
Updating cb2e987..ef759b0
Fast-forward
 README.md                    | 25 +++++
 app/app.css                  | 58 ++++++---
 app/app.html                 | 3 +-
 datos/datos.html            | 6 +-
 .../Casos de uso del Proyecto de Software.pdf | Bin 0 -> 131357 bytes
 other-data/Proyecto de Software_JSS.pdf       | Bin 0 -> 920775 bytes
 resultados/resultados.html  | 125 ++++++-----
 7 files changed, 169 insertions(+), 48 deletions(-)
 create mode 100644 README.md
 create mode 100644 other-data/Casos de uso del Proyecto de Software.pdf
 create mode 100644 other-data/Proyecto de Software_JSS.pdf
```

```
MINGW64/c/Users/jaime/OneDrive/Imágenes/Escritorio/PrototipoDataBloo...
jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git add .

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git commit -m"Version 0.1.1 - Creacion Clase Java"
[master b715a34] Version 0.1.1 - Creacion Clase Java
 1 file changed, 66 insertions(+)
 create mode 100644 ReporteSoporte.java

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git push -u jaimee
fatal: 'jaimee' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (master)
$ git push -u jaime
```

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 845 bytes | 845.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/JaimeSanchez75/DataBloomAnalytics.git
   ef759b0..b715a34  master -> master
branch 'master' set up to track 'jaime/master'.
```

```
jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (
master)
$git checkout -b pruebas
Switched to a new branch 'pruebas'

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (
pruebas)
$ git add .

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (
pruebas)
$ git commit -m"Version 0.1.2 - Caja negra y blanca"
[pruebas ac88e3c] Version 0.1.2 - Caja negra y blanca
 2 files changed, 25 insertions(+)
 create mode 100644 "cajaBlanca/C\303\263digo M\303\263dulo Proyecto.pdf"
 create mode 100644 cajaNegra/TestJava.java

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (
```

```
pruebas)
$ git push jaime pruebas
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 124.25 KiB | 24.85 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'pruebas' on GitHub by visiting:
remote:   https://github.com/JaimeSanchez75/DataBloomAnalytics/pull/new/prueb
as
remote:
To https://github.com/JaimeSanchez75/DataBloomAnalytics.git
 * [new branch]      pruebas -> pruebas

jaime@Jaime MINGW64 ~/OneDrive/Imágenes/Escritorio/PrototipoDataBloomAnalytics (
pruebas)
$ git checkout -b proyecto
Switched to a new branch 'proyecto'
```

