



PROYECTO BASE DE DATOS && PROGRAMACIÓN

Aimar Esqueta Albert

José Manuel Piñero Ávila

VAARI APP



Índice

Descripción general del proyecto y objetivos	3
Instrucciones de instalación y configuración del entorno de desarrollo y la base de datos	7
Guía de uso de la aplicación, con capturas de pantallas y ejemplos de interacción	12
Descripción de las principales funcionalidades implementadas y tecnologías utilizadas	23
Explicación del diseño de la base de datos	25
Sección de Conclusiones	28



Descripción general del proyecto y objetivos

¿Qué es VAARI?

VAARI es una app para la gestión de unidades de día de ancianos. Además de ser una app que ayuda a la gestión de los voluntarios y trabajadores para un mejor cuidado de nuestros mayores, VAARI a su vez es una agenda digital y una app de comunicación en la que podremos contar con diferentes funcionalidades para siempre estar en contacto con familiares y trabajadores para que nuestros mayores se sientan siempre arropados y cuidados.

Para nosotros no hay nada más bonito que mantener la sonrisa de nuestros mayores y sus familias. Por eso hemos venido como una app que revolucionará el sector de gestión en este tipo de centros.

Nuestras funcionalidades

La aplicación implementará en sí misma **dos tipos de acceso** a la app: un perfil de usuario para los familiares y otro perfil de funcionario para los empleados.

-Fichas y registro de nuestros mayores exactas

Los empleados tendrán en su perfil el poder de editar y registrar a los ancianos, con sus datos personales (descripción de trastorno , personalidad y actitudes de forma cercana, gustos y cosas que le hacen bien etc..).

Toda esa descripción a ser posible hecha por parte del familiar para que sea cercana y exacta.



-Seguimiento de las comidas de nuestros mayores, su alimentación, durante la estancia en la unidad de día. (Desayuno, Almuerzo, Cena si la hubiese), interactiva y visual con estados como (Bueno, bien, poco, completo, etc..) con colores según el estado.

o bien, separado por menú.

-Calendario de eventos/actividades para que los familiares estén al tanto de todas las actividades que harán nuestros mayores durante su estancia en el centro.

-Cuidados exhaustivos para los más necesitados

Control de heces, control de sueño, control de medicación para nuestros ancianos con demencia senil o enfermedades degenerativas.

-Recordatorios de las tomas por alarma a los funcionarios de cada anciano.

-Control de horarios

Hora de entrada y salida del centro de día

Avisos por la app en modo de difusión.

-Una sección especial para la auxiliar de enfermería de ayuda a domicilio

Para el registro de ancianos. Si existiese algún familiar vinculado, poder informarle de todo igualmente.

-Chat bot que devuelve información de datos solicitados

¿Por qué VAARI?

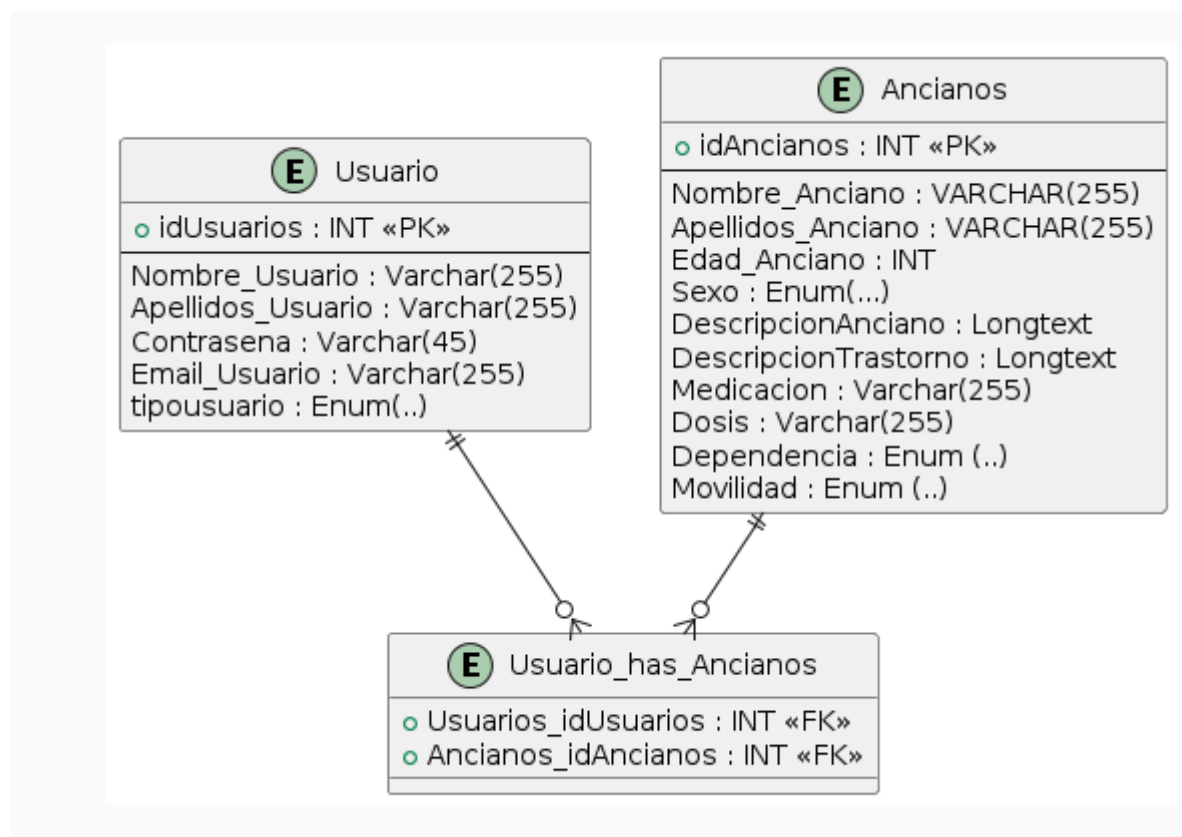
Queremos aportar nuestro granito de arena para que la estancia de nuestros mayores en centros de día sea siempre la mejor, por eso seremos el intermediario perfecto entre centros y familias.

Por todos esos abuelos que cuidaron de nosotros como padres...

¡¡¡Ahora nos toca a nosotros devolverles ese cariño!!!

Nota: Algunas de las funcionalidades podrían verse modificadas, eliminadas o añadidas algunas otras.

Diagrama del Modelo de Datos: Un diagrama visual del modelo de datos utilizado, incluyendo entidades, relaciones, y una breve descripción de cada tabla y campo





Breve descripción de cada tabla y campo

Tabla: USER

idUsuarios: Identificador único del usuario.

Nombre_Usuario: Nombre del usuario.

Apellidos_Usuario: Apellidos del usuario.

Contraseña: Contraseña del usuario.

Email_Usuario: Dirección de correo electrónico del usuario.

tipousuario: Tipo de usuario, puede ser empleado, familiar u otro.

Tabla: ANCIANOS

idAncianos: Identificador único del anciano.

Nombre_Anciano: Nombre del anciano.

Apellidos_Anciano: Apellidos del anciano.

Edad_Anciano: Edad del anciano.

Sexo: Género del anciano, puede ser masculino o femenino.

DescripcionAnciano: Descripción del anciano, incluyendo información adicional sobre su personalidad, gustos, etc.

DescripcionTrastorno: Descripción de los trastornos o enfermedades del anciano, si los hubiera.

Medicacion: Medicación que toma el anciano.

Dosis: Dosis de la medicación.

Dependencia: Nivel de dependencia del anciano.



Movilidad: Nivel de movilidad del anciano.

Tabla: USER_HAS_ANCIANOS

Usuarios_idUsuarios: Identificador del usuario relacionado.

Ancianos_idAncianos: Identificador del anciano relacionado.

Instrucciones de instalación y configuración del entorno de desarrollo y la base de datos

En este punto queremos explicar la serie de instrucciones que hemos seguido para la configuración de los entornos de desarrollo que hemos utilizado y la base de datos.

Queremos explicar punto por punto lo que hemos hecho desde el minuto 0 para que quede constancia de todo el proceso.

Instalación y Configuración del Entorno de Desarrollo:

1. Docker:



Optamos por Docker para garantizar un entorno de desarrollo consistente y portátil.

Instalamos Docker Desktop siguiendo las instrucciones del sitio web oficial para nuestros respectivos sistemas operativos.

Además, utilizamos la carpeta de Docker LAMP que incluye contenedores Docker preconfigurados para cada uno de estos componentes (PHP, MySQL, Apache, Linux), lo que



facilita la creación rápida y la ejecución de un entorno de desarrollo completo en cualquier sistema operativo que admita Docker.

En nuestro caso, hemos utilizado MySQL el cual es un sistema de gestión de bases de datos relacional de código abierto.

El contenedor Docker MySQL en Docker LAMP proporciona un servidor MySQL preconfigurado que puede almacenar y administrar los datos de las aplicaciones web.

2. MySQL Workbench:



Descargamos e Instalamos MySQL Workbench:

Ejecutamos el instalador descargado y seguimos las instrucciones del asistente de instalación para instalar MySQL Workbench en nuestro sistema. Durante el proceso de instalación, aseguramos que se incluyera el servidor MySQL para tener una instancia local disponible en nuestra máquina.

Configuración Inicial:

Después de la instalación, abrimos MySQL Workbench y seguimos los pasos para realizar la configuración inicial. Esto puede incluir la configuración de la conexión a una instancia local de MySQL, proporcionando credenciales de acceso y configurando otras opciones según nuestras necesidades.

Conexión a la Base de Datos:

Una vez que MySQL Workbench está configurado, podemos conectarnos a la instancia local de MySQL o a cualquier otra instancia remota a la que necesitemos acceder. Proporcionamos la información de conexión necesaria, como la dirección IP, el nombre de usuario y la contraseña.



Creación y Administración de Bases de Datos:

Utilizamos MySQL Workbench para crear y administrar nuestras bases de datos MySQL aprovechándonos de su interfaz gráfica intuitiva para la creación de modelos relacionales. Esto incluye la creación de nuevas bases de datos, tablas, consultas SQL, procedimientos almacenados, y otras tareas de administración de bases de datos.

Al elegir MySQL y MySQL Workbench, nos beneficiamos de la fiabilidad y la amplia adopción de MySQL como sistema de gestión de bases de datos, así como de la comodidad y la funcionalidad que ofrece MySQL Workbench como herramienta de administración. Esto nos ha permitido diseñar, desarrollar y administrar eficientemente nuestra base de datos para la aplicación VAARI.




3. Eclipse:



Para la creación de este proyecto hemos utilizado Eclipse ya que es el IDE con el que nos hemos sentido más cómodos durante el curso. Para ello hemos configurado el entorno para facilitarnos su uso.

Para conectar nuestro proyecto Eclipse con la base de datos MySQL, utilizamos un archivo JAR que contenía el controlador JDBC necesario para establecer la conexión. Descargamos este archivo JAR y lo incluimos en el directorio de nuestro proyecto.

La clase ConexionMySQL y el archivo JAR los importamos del anexo que se nos proporcionó en el módulo de programación.

 ConexionMySQL	02/06/2020 13:20	Archivo de origen ...	4 KB
 datos	02/06/2020 12:26	Archivo de origen ...	1 KB
 mysql-connector-java-8.0.20	09/03/2020 11:19	Archivo JAR	2.330 KB



La clase ConexionMySQL que contiene la lógica para establecer y gestionar la conexión con la base de datos MySQL. Esta clase se encargaba de cargar el controlador JDBC, establecer la conexión con la base de datos utilizando la información de conexión proporcionada y proporcionar métodos para ejecutar consultas SQL y realizar otras operaciones en la base de datos.

Dentro de nuestro proyecto Eclipse, utilizamos la clase ConexionMySQL para conectarnos a la base de datos MySQL y realizar operaciones como consultas, inserciones, actualizaciones y eliminaciones de datos.

Configuramos la conexión a la base de datos utilizando la información de conexión adecuada, como la URL de la base de datos, el nombre de usuario y la contraseña, según fuera necesario para nuestro entorno de desarrollo.

Al utilizar Eclipse como nuestro IDE principal y configurarlo para trabajar con MySQL, pudimos desarrollar y mantener nuestro proyecto de manera eficiente, aprovechando las funcionalidades y herramientas proporcionadas por Eclipse y la facilidad de conexión a la base de datos.

Instrucciones de instalación y configuración de la base de datos.

La configuración de la base de datos para nuestro proyecto VAARI implicó los siguientes pasos:

Creación de la Base de Datos:

Utilizamos MySQL Workbench para crear una nueva base de datos llamada freedb_VAARI_APP. Esta base de datos sería el almacén principal de todos los datos relacionados con nuestra aplicación VAARI.

Definimos la estructura de la base de datos diseñando un conjunto de tablas que representan las entidades principales de nuestra aplicación, como Ancianos y Usuarios. Estas tablas fueron diseñadas teniendo en cuenta los requisitos funcionales y la lógica de negocio de la aplicación.



Importación del Esquema:

Importamos el esquema de la base de datos utilizando un script SQL proporcionado previamente en MySQL Workbench. Este script SQL contenía las instrucciones necesarias para crear las tablas y definir las relaciones entre ellas.

Esta importación automatizó el proceso de creación de las tablas y aseguró que la base de datos estuviera correctamente configurada según los requisitos de nuestra aplicación VAARI.

Conexión de Eclipse con la Base de Datos:

Configuramos la URL de conexión, el nombre de usuario y la contraseña correspondientes en la clase ConexionMySQL, proporcionando así los detalles necesarios para establecer la conexión con la base de datos desde nuestra aplicación Java.

Estos pasos nos permitieron configurar eficazmente la base de datos para nuestro proyecto VAARI y establecer una conexión funcional entre nuestra aplicación Java en Eclipse y la base de datos MySQL.

Guía de uso de la aplicación, con capturas de pantalla y ejemplos de interacción.

Primeros Pasos en la APP

Registro de Usuario

En Nuestra App el primer paso que debes de seguir para poder acceder a las funcionalidades que ofrecemos a la hora de entregar este proyecto y en visión de futuro, es la de registrar tu usuario.

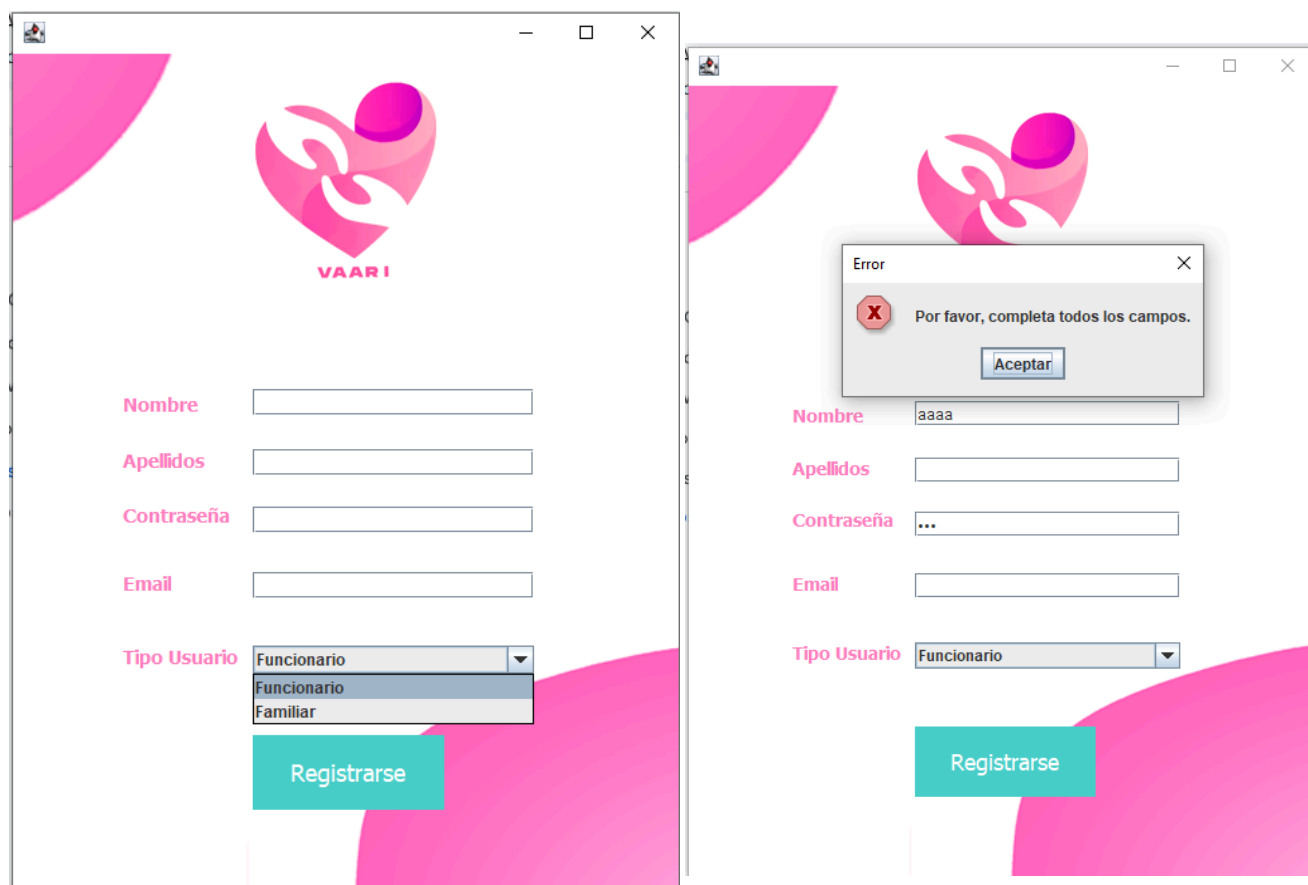
En esta sección tendrás que dejarnos datos como :

-Tu Nombre

-Tus Apellidos

-Tu Email

-Y seleccionar el tipo de Usuario que vas a ser. En el caso de ser familiar de un miembro hospedado en la estancia diurna en cualquiera de nuestros centros de día o perfil funcionario en el caso que seas trabajador de cualquiera de los centros en los que estamos presentes.



The image displays two screenshots of a web application's registration form. The form is titled 'VAARI' and includes fields for 'Nombre', 'Apellidos', 'Contraseña', 'Email', and 'Tipo Usuario'. The 'Tipo Usuario' dropdown menu is open, showing options for 'Funcionario' and 'Familiar'. A 'Registrarse' button is located at the bottom of the form. The right screenshot shows the same form with an error message: 'Error: Por favor, completa todos los campos.' and an 'Aceptar' button.

En el caso que hubiese algún error o campo de texto inválido o vacío la APP te devolverá una ventana de mensaje emergente como error , avisando de ello para que sepas dónde fallaste.

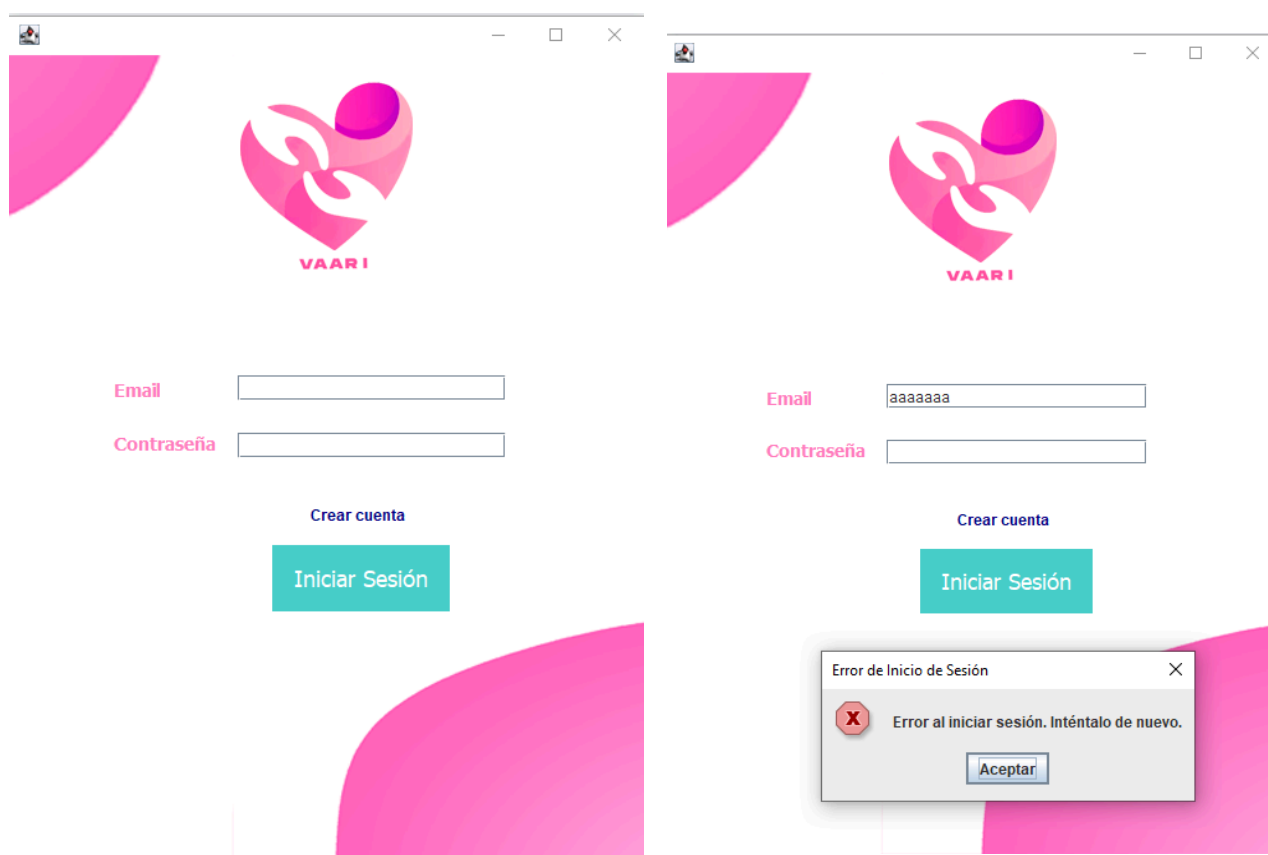
Inicio de Sesión

Seguido de todo esto y si todo está correcto pasarás al siguiente paso en la APP.

El inicio de sesión con tu Email en este caso y tu contraseña.

Dentro de esta interfaz deberás escribir tu Email y contraseña a tu cuenta VAARI .

Si existiese algún error aparecerá un mensaje de error en ventana emergente indicando de ello.



Una vez iniciada la sesión aparecerá la página principal.

Visualmente y a nivel funcional solo es una ventana que funciona para hacerte ver que todo está bien y estás dentro de la APP con tu usuario logueado de forma correcta.

A futuro pondremos accesos, vistas rápidas a actividades recientes .



Vista de Página Principal

Dentro de la Página principal hemos programado un menú desplegable en la parte superior izquierda en el que te encontrarás un marcador con la fecha actual del día de hoy.

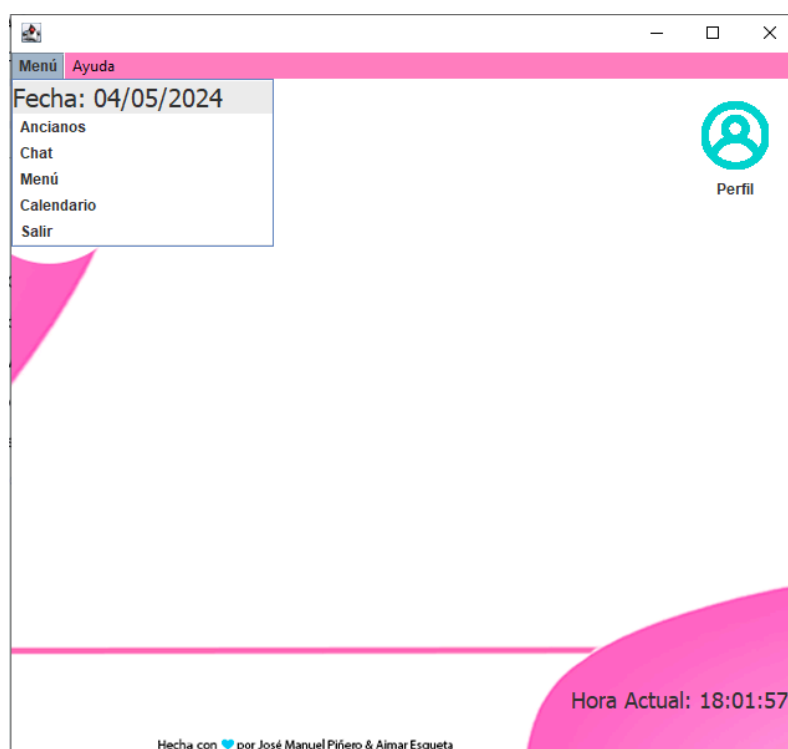
Seguido de :

-Sección Ancianos en las que el funcionario podrá crear y registrar un nuevo miembro en su equipo o consultar el registro de su grupo. (Tenemos que restringir el acceso a esta al usuario Familiar aún, y darle otra perspectiva distinta, en la que podrá ver los datos de su Familiar miembro de un centro que haga uso de VAARI).

-Sección calendario en las que ambos roles de usuario van a poder ver eventos o dejar notas.

-Sección chat, en la que un chatbot te ayudará en cualquier pequeña consulta(esta sección aún sigue en desarrollo pero nos ha gustado mucho la idea de comenzar a implementarla para tener una visión de cómo será)

-Salir, que te desconectará como usuario y cerrará tu sesión .



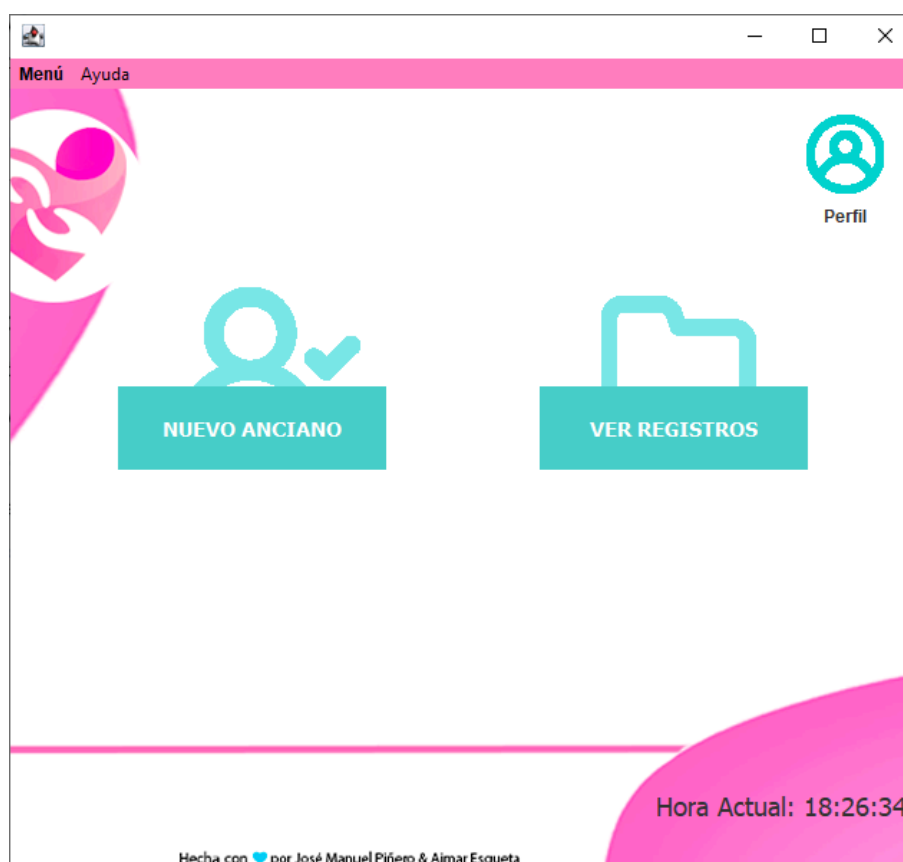


En la pantalla principal hemos programado un reloj en vivo.

También hemos comenzado a implementar la opción perfil. Al pulsar un botón interactivo , sobre el logo de Usuario podrás acceder a una mini ventana en la que gracias a una consulta de tus datos de usuario de la base de datos logueados en ese momento, podrás ver de vuelta tu Nombre de Usuario (Email) y Apellidos con tu Rol de usuario.

Sección Ancianos

Dentro de la sección Ancianos como bien hemos comentado , el funcionario verá dos botones interactivos. Uno le dará acceso a una ventana para crear un nuevo registro de miembro de su grupo de pacientes al rellenar los campos necesarios con los atributos que le corresponden. Y otro le dará acceso al escribir el Nombre y Apellidos del anciano que precise en ese momento obtener su información o ficha completa.



Una vez que haga clic en nuevo anciano le aparecerá una nueva ventana para crear el nuevo registro del paciente.



Me alegra tenerte por aquí...

Y claro que nos alegra tenerte por aquí, hoy nuestro trabajo da un nuevo fruto y es tenerte aquí. Para velar por tu bienestar junto con todo el equipo humano que va a rodearte.

Nuevo Registro

Una vez dentro , el funcionario tiene que rellenar los campos requeridos del nuevo Anciano Miembro de su grupo. Atributos como :

-Nombre

-Apellidos

-Sexo

-Edad

-Gustos y hobbies : en esta campo queremos darle voz o personalidad a aquellas personas sobre todo que resultan olvidadas socialmente incluso lamentablemente en las propias unidades de días o residencias. Los enfermos de alzheimer o de demencia. Es por ello que nació este proyecto, por ellos y por sus familiares. Queremos darle voz y recuerdo a lo que alguna vez les dió vida, momentos de felicidad y les hizo sentir vivos y con lucidez plena .

Queremos saber que A Josefa le encanta el Flamenco, escuchar a su Juanito Valderrama con esa voz tan española y peculiar cantando su " El Emigrante" .

Queremos y necesitamos saber también que a Enrique le encanta y no podía ni puede vivir sin su Betis .

Queremos saber que a Margarita le encanta el campo y le brilla la cara hablar con la gente.

Queremos saber que Isabel trabajó en el campo cogiendo algodones para alimentar a su familia de 5 hijos , que vivió en Alemania por trabajo y de siempre una buena sevillana y juerga le devuelve un ratito de lucidez.



-Trastornos o enfermedad : en este campo el funcionario tendrá que especificar trastornos o enfermedades que tenga el paciente.

-Medicación: si la hubiese el funcionario rellenará con los nombres de las mismas y si no , con un no bastará.

-Dosis: aquí se indicarán en números las dosis de cada medicación.

-Dependencia: aquí se indicará el grado de dependencia del paciente.

-Movilidad: aquí se indicará el grado de movilidad del paciente

Una vez completado todos estos campos, tenemos un botón interactivo CREAR , este rellenará y creará un nuevo paciente en nuestra base de datos, quedando registrado en ella hasta que sea modificado o borrado a futuro.

Menú Ayuda

Me alegra tenerte por aquí...

Nombre Sexo

Apellidos

Edad

Gustos/Hobbies ¿Como es...?

Trastornos o Enfermedad

Medicación Dosis

Dependencia Movilidad

CREAR

Hora Actual: 18:29:28

Si volvemos atrás o una vez creado el Registro, nos aparecerá de nuevo la ventana principal de la sección ancianos. El funcionario podrá como hemos dicho hacer una consulta en la base de datos con el botón Ver registro.

Una vez dentro nos aparecerá esta ventana, que rellenando Nombre y Apellidos de anciano nos generará un archivo .txt que contendrá todos los datos antes escritos en su registro o en cualquier etapa en el tiempo.



Menú Ayuda

Me alegra tenerte por aquí...

Nombre

Apellidos

DESCARGAR FICHA

Hora Actual: 18:46:38

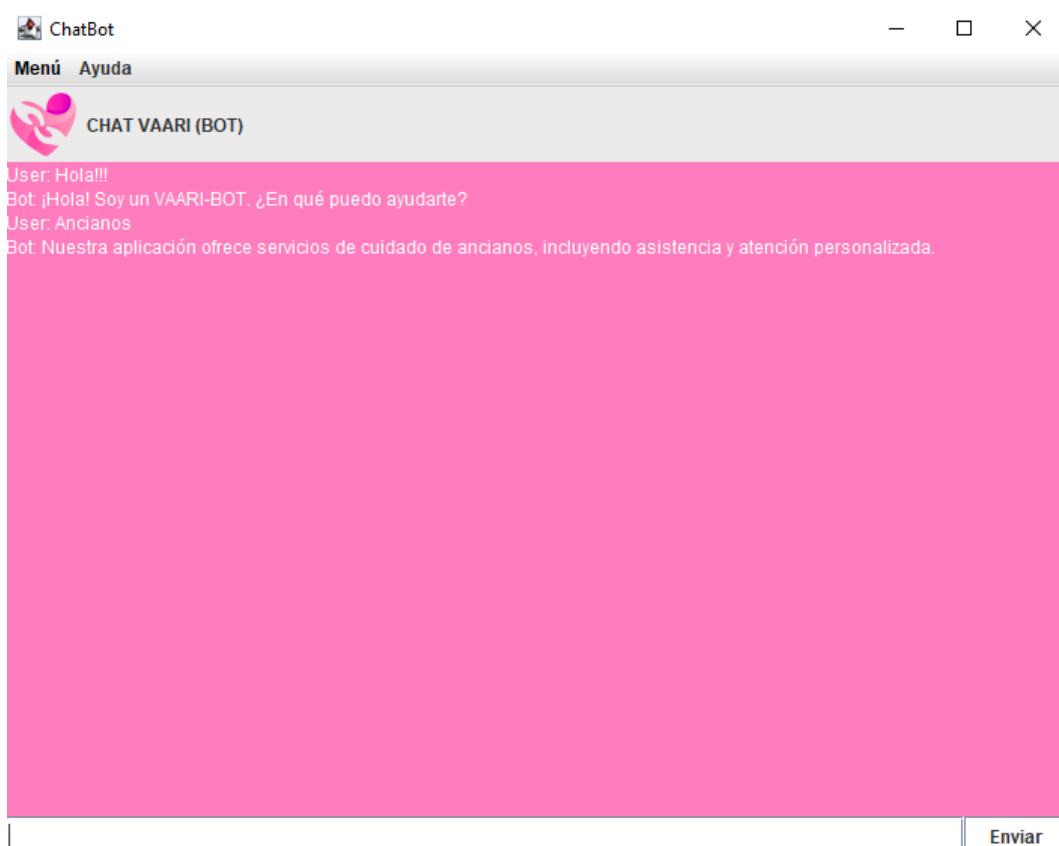
Volvemos al menú desplegable y navegaremos hasta el chatBot

ChatBot

Hemos implementado un chatBot y seguiremos con ello. Este chatBot ayudará tanto al funcionario como al Familiar en pequeñas consultas escribiendo palabras claves, que el bot reconocerá y devolverá en la caja de texto de chat los datos más precisos para que la persona que lo use no necesite ayuda más compleja. En el caso de que eso último se diese siempre el chatBot dará la opción de contacto por teléfono con el centro (Ayuda a familiar) o

contacto con soporte de VAARI si se tratase de un error en la App ante el uso de cualquier funcionario .

Hemos programado alguna de esas palabras claves con un pequeño mensaje de vuelta de nuestro Bot para que se pueda ver un poco la funcionalidad real.



Calendario

Estamos dentro del Calendario de la APP, Día, fecha y hora están aquí presentes.

Dividido por los días del mes puedes dejar alguna nota marcada en cualquier día del mes.

(Aún no se quedan guardadas en el usuario si éste cierra la ventana o cierra su sesión, estamos implementando esa opción con una tabla en base de datos que relacione con el usuario)



Ambos roles de usuario tienen acceso a esta función. Una vez seleccionen el día hemos programado una ventana de mensaje en la que escribirás la nota de texto o recordatorio para ese día para registrarla.

Calendario

Menú Ayuda

Calendario

Fecha y Hora Actual: 04/05/2024 19:18:57

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Añadir Nota

Ingresa una nota para el día 2

Ir de visita

Aceptar Cancelar

Calendario

Menú Ayuda

Calendario

Fecha y Hora Actual: 04/05/2024 19:19:07

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Nota para el día 2: Ir de visita



Volviendo a la Pantalla principal junto al Menú desplegable que hemos estado documentando y explorando tenemos un menú desplegable de Ayuda.

Este presente y usable para cualquier Rol de Usuario.

Dentro de este menú hemos incluido...

Menú Ayuda

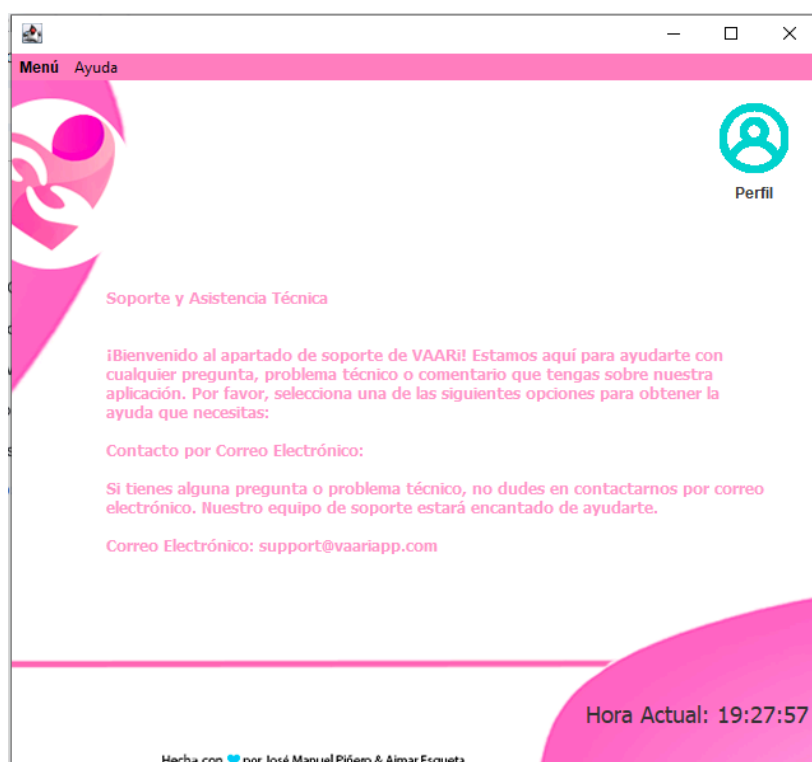
-Sobre VAARI



Sección en la que hemos descrito el proyecto y un poco nuestros valores o filosofía.

-Soporte

Sección en la que podrás encontrar nuestro correo electrónico de soporte ante cualquier duda, soporte técnico o suscribirte a la newsletter para estar al tanto de actualizaciones y agradecimientos.



-Guía Una pequeña guía para que el funcionario explique al familiar como usar nuestra aplicación y que ofrecemos. O que sirva de guía para un nuevo funcionario que comience a disfrutar de nuestros servicios.





Descripción de las principales funcionalidades implementadas y tecnologías utilizadas

Funcionalidades Principales Implementadas:

Gestión de Ancianos:

Registro y edición de ancianos con datos personales detallados.

Seguimiento de la alimentación y estado de salud durante su estancia.

Control exhaustivo de aspectos como la medicación, sueño y movilidad.

Calendario de Eventos/Actividades:

Agenda digital para mantener a los familiares informados sobre las actividades programadas.

Control de Horarios:

Registro de entradas y salidas del centro de día.

Notificaciones por la aplicación para difundir avisos importantes.

Chat Bot:

Asistente virtual para proporcionar información y ayuda rápida a los usuarios.

Tecnologías Utilizadas:

Desarrollo de la Aplicación:

Utilización de Eclipse como IDE principal para el desarrollo.

Implementación en Java para la lógica de la aplicación.



Gestión de Base de Datos:

Uso de MySQL como sistema de gestión de bases de datos.

Utilización de MySQL Workbench para el diseño y administración de la base de datos.

Entorno de Desarrollo:

Docker para garantizar un entorno de desarrollo consistente y portátil.

Configuración de contenedores Docker LAMP para PHP, MySQL, Apache, Linux.

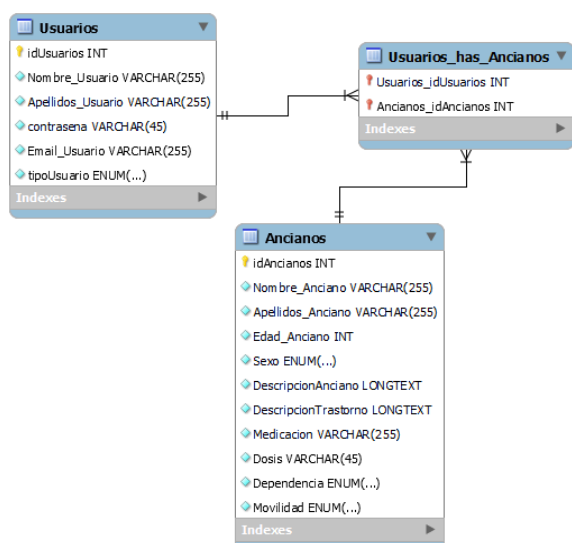
Funcionalidad	Descripción
Gestión de Ancianos	Registro y edición de ancianos con datos personales detallados, seguimiento de alimentación y salud, control exhaustivo de aspectos como la medicación y movilidad.
Calendario de Eventos/Actividades	Agenda digital para informar a los familiares sobre actividades programadas en el centro de día.
Control de Horarios	Registro de entradas y salidas del centro de día, notificaciones por la aplicación para difundir avisos importantes.
Chat Bot	Asistente virtual para proporcionar información y ayuda rápida a los usuarios.
Eclipse	IDE utilizado para el desarrollo de la aplicación.
Java	Lenguaje de programación utilizado para la lógica de la aplicación.



MySQL	Sistema de gestión de bases de datos utilizado para almacenar y administrar datos de la aplicación.
MySQL Workbench	Herramienta utilizada para el diseño y administración de la base de datos MySQL.
Docker	Utilizado para garantizar un entorno de desarrollo consistente y portátil.
Contenedores Docker LAMP	Configuración de contenedores para PHP, MySQL, Apache, y Linux para el entorno de desarrollo.

Explicación del diseño de la base de datos, incluyendo el modelo de datos

El diseño de nuestra base de datos freedb_VAARI_APP se ha realizado de manera cuidadosa para garantizar la eficiencia, la integridad de los datos y la escalabilidad del sistema.





A continuación, explicaremos cada componente del diseño:

Tablas Principales:

Ancianos: Esta tabla almacena información detallada sobre los ancianos atendidos en la aplicación. Cada anciano tiene un identificador único (idAncianos) y se registran datos como nombre, apellidos, edad, género, descripciones sobre el anciano y sus trastornos, medicación, dosis, nivel de dependencia y movilidad.

Usuarios: Aquí se registran los usuarios que interactúan con la aplicación, distinguiendo entre funcionarios y familiares. Cada usuario tiene un identificador único (idUsuarios) y se almacenan datos como nombre, apellidos, contraseña, dirección de correo electrónico y tipo de usuario.

Relación Muchos a Muchos:

Usuarios has Ancianos: Esta tabla establece una relación muchos a muchos entre usuarios y ancianos. Esto significa que un usuario puede estar asociado con varios ancianos, y a su vez, un anciano puede ser atendido por varios usuarios. Esta estructura proporciona flexibilidad y permite que diferentes usuarios accedan y gestionen la información de los ancianos.

Claves Primarias y Secundarias:

Cada tabla tiene una clave primaria (idAncianos para Ancianos, idUsuarios para Usuarios) que garantiza la unicidad de cada fila.

En la tabla Usuarios_has_Ancianos, las claves primarias son combinaciones de Usuarios_idUsuarios y Ancianos_idAncianos, lo que asegura que no se pueda duplicar la asociación entre usuarios y ancianos.



Tipos de Datos y Restricciones:

Se han utilizado tipos de datos apropiados para cada campo, como VARCHAR para cadenas de texto y ENUM para campos con valores predefinidos como género, nivel de dependencia, etc.

Se han añadido restricciones de clave externa para garantizar la integridad referencial entre las tablas, lo que significa que no se pueden insertar filas que hagan referencia a valores inexistentes en otras tablas.

Normalización:

Se ha seguido el principio de normalización para reducir la redundancia y evitar anomalías en los datos. Cada tabla almacena información específica y no se repite la misma información en diferentes tablas.

Ahora vamos a explicar un poco las decisiones que hemos tomado para llegar a la base de datos relacional de VAARI que hemos creado para garantizar su eficiencia, integridad y escalabilidad.

Aquí, contaremos las decisiones claves por las que hemos optado para la creación del diseño final de nuestra BBDD:

Normalización de datos: Lo que significa que se evita la redundancia de datos y se promueve la integridad de los datos. Esto facilita la actualización y modificación de los datos sin introducir inconsistencias.

Uso de claves primarias y foráneas: El uso de claves primarias y foráneas garantiza la integridad referencial en la base de datos. Esto ayuda a mantener la coherencia de los datos al establecer relaciones entre las tablas y evitar la inserción de registros huérfanos.

Tipos de datos adecuados: Se han elegido tipos de datos adecuados para cada columna, lo que permite almacenar y manipular los datos de manera eficiente. Por ejemplo, se utilizan



VARCHAR para campos de texto de longitud variable y ENUM para campos con un conjunto fijo de valores.

Motor de almacenamiento InnoDB: El uso del motor de almacenamiento InnoDB es una elección común para bases de datos que requieren transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad). Esto garantiza la integridad de los datos incluso en situaciones de concurrencia.

Relaciones entre tablas: Las relaciones entre las tablas (por ejemplo, entre Usuarios y Ancianos en la tabla Usuarios_has_Ancianos) permiten modelar la asociación entre entidades de manera efectiva. Esto facilita consultas complejas y el mantenimiento de la consistencia de los datos.

Autoincremento en claves primarias: El uso de la característica AUTO_INCREMENT en las claves primarias simplifica la inserción de datos al permitir que la base de datos genere automáticamente valores únicos para estas columnas. Esto reduce la carga sobre el desarrollador y evita conflictos de identidad.

En resumen, nuestras decisiones de diseño tomadas están alineadas con las mejores prácticas de diseño de bases de datos relacionales y están destinadas a garantizar la eficiencia, integridad y flexibilidad de la base de datos de VAARI.

Conclusiones

Podemos decir que el proceso de creación de esta aplicación ha sido una experiencia emocionante y gratificante. Desde la concepción de la idea hasta la implementación final, cada paso ha sido un desafío que hemos abordado con determinación y creatividad.

Optamos por utilizar tecnologías como Docker, MySQL Workbench y Eclipse para garantizar un desarrollo eficiente y colaborativo. La configuración del entorno de desarrollo, la creación de la base de datos y la conexión con la aplicación fueron aspectos clave que abordamos con cuidado y atención a los detalles.

Durante el proceso, enfrentamos desafíos como la configuración inicial del entorno y el diseño del modelo de datos, pero logramos superarlos con trabajo en equipo y perseverancia. Así como encontramos e mayor desafío en la conexión entre nuestro IDE y la base de datos, finalmente conseguimos solucionar todos estos problemas. Aprendimos la importancia de planificar y diseñar adecuadamente antes de comenzar el desarrollo, así como la necesidad de ser flexibles y adaptativos a medida que surgían nuevos requisitos y desafíos.

En un futuro nos encantaría seguir mejorando la app y poder contar con nuevas funcionalidades como:

- *La mejora de la interfaz ver perfil:* Hemos creído conveniente empezar ya algunas nuevas interfaces y funcionalidades que iremos aportando en nuevas actualización de la app, aún no está disponible ver perfil por lo que ahora mismo solo podremos ver una foto del usuario o un icono reemplazando si no hubiese foto.
- La idea de esta interfaz es que vaya haga una consulta a la base de datos y nos devuelva todos los datos de nuestro usuario en modo de desplegable



Menú Ayuda

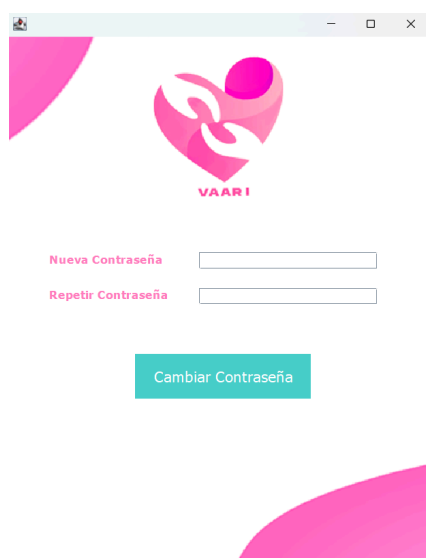
Tu Perfil

Usuario

Apellidos

Tipo Usuario

- *Recordar contraseña:* Clase que se implementara en un futuro para que los usuarios puedan recuperar sus claves aún no tenemos esta funcionalidad activada ya que nos haría falta poder mandar correos de confirmación a los diferentes emails de los usuarios pero una futura actualización de la app llegará



En resumen, VAARI es el resultado de un arduo trabajo, colaboración y pasión por crear una aplicación que mejore la calidad de vida de las personas mayores y sus cuidadores. Estamos emocionados de ver cómo VAARI crece y evoluciona en el futuro, y estamos comprometidos a seguir mejorando y expandiendo esta plataforma para hacer una diferencia positiva en la vida de los usuarios.

