



"Match, code, create"

Gabriel Carmona Palop

Alba Sierra Vives

Adrián Pons Chovi

Hugo Miquel Valles



Indice

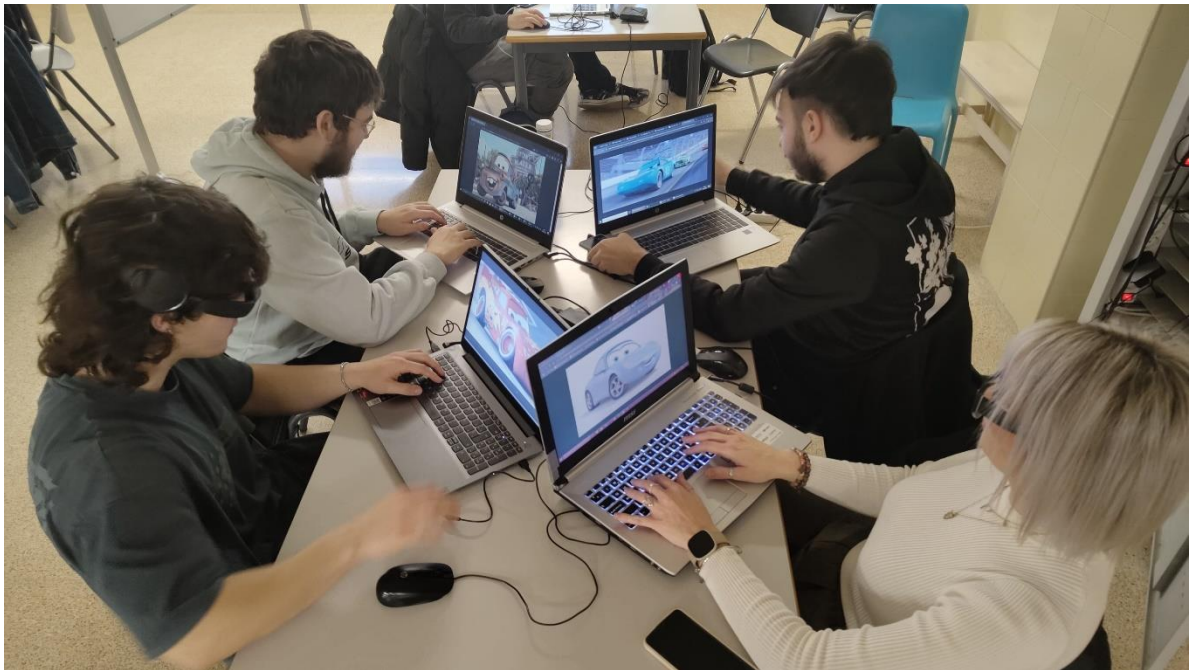
Introducción	3
HCM	4
Estudio de Viabilidad económica, humana y técnica	7
Plan de empresa	11
Términos y Condiciones	16
Política de Protección de Datos de Love4Code	18
Daily Scrum Meetings	20
Documentación BBDD	25
Documentación API	28
Pantallas APP	33
Futuras Mejoras	45
Conclusión	45



Introducción

Somos un equipo de cuatro estudiantes de 2º de Desarrollo de Aplicaciones Multiplataforma (DAM), y hemos desarrollado "Love 4 Code", una aplicación móvil que combina una red social profesional con una plataforma de citas, dirigida especialmente a programadores. La app permite a los usuarios conectar con otros profesionales del sector tecnológico a través de un sistema de swipe, donde pueden dar "like" o "dislike" a perfiles basados en habilidades, experiencia y tecnologías que manejan. Además, incluye funcionalidades como chats en tiempo real y personalización de perfiles.

Este proyecto es el resultado de un trabajo académico realizado en un sprint de dos semanas, donde hemos aplicado los conocimientos adquiridos en el ciclo formativo para crear una aplicación funcional y escalable. "Love 4 Code" no solo es una app, sino también una muestra de nuestro aprendizaje y esfuerzo como futuros desarrolladores.





Hugo:

EMPLEADO

Empleados

codigo empleado:

#

2

email:

humiva@alumnatflorida.es

phone:

622923562

nombre:

T Hugo

apellidos:

T Miquel Valles

imagen:

Gabriel:

EMPLEADO

Empleados

codigo empleado:

#

3

email:

gacapa@alumnatflorida.es

phone:

644365817

nombre:

T Gabriel

apellidos:

T Carmona Palop

imagen:

Adrián:

EMPLEADO

Empleados

codigo empleado:

#

4

email:

adpoch@alumnatflorida.es

phone:

622230884

nombre:

T Adrián

apellidos:

T Pons Chovi

imagen:

5

Love
4
Code

Tablas proyectos y tareas, con los empleados asignados a cada proyecto y cada tarea:

PROYECTOS



	Nombre Empleado	Nombre Proyecto	Fecha Entrega	Presupuesto Estimado
	Adrián	proyecto01	12/02/2025	20.05
	Alba	proyecto02	19/02/2025	10
	Gabriel	proyecto03	28/02/2025	50
	Hugo	proyecto04	26/02/2025	47.99

LISTADO DE TAREAS



Nombre Empleado	Nombre Proyecto	Nombre Tarea	Tiempo Realizacion	Bonificacion
Alba	proyecto01	florida expo	2	3
Adrián	proyecto02	backend	10	50
Hugo	proyecto03	frontend	6	45
Gabriel	proyecto04	full stack	12	80



Estudio de Viabilidad económica, humana y técnica

1. Introducción

Nuestro equipo de 4 estudiantes de programación ha propuesto desarrollar "Love 4 Code", una aplicación móvil que mezcla lo mejor de una app de citas con una red social profesional enfocada en programadores. La idea es que los usuarios hagan swipe en perfiles de otros programadores según sus habilidades, experiencia y tecnologías que manejan. El objetivo es conectar a personas con intereses similares en la programación, ya sea para trabajo, colaboración en proyectos o incluso algo más.

El desarrollo de la app se llevará a cabo en un sprint de 2 semanas, dentro del horario de clases. En este estudio analizamos la viabilidad del proyecto en términos económicos, humanos y técnicos.

2. Viabilidad Económica

Presupuesto disponible: 0€. Contamos solo con recursos gratuitos y lo que nos proporcionen en Florida.

Infraestructura: Podemos usar AWS para hosting, GitHub para control de versiones, y plataformas de desarrollo gratuitas como Eclipse y Expo Snack.

Costes de publicación: Subir la app a Google Play tiene un costo de 25\$, pero como es un proyecto académico, podemos distribuirla en formato APK o usar TestFlight para pruebas en iOS.

Alternativa: Si queremos lanzar la app a más personas sin costos extra, podemos hacer una versión web con React y alojarla en AWS.

Conclusión económica: Viable si usamos solo herramientas gratuitas y no buscamos lanzar la app en tiendas oficiales en esta fase.



3. Viabilidad Humana

Equipo:

- **Desarrollador 1 (Adrián):** Backend y base de datos.
- **Desarrollador 2 (Hugo):** Frontend y UI/UX.
- **Desarrollador 3 (Gabriel):** Backend y base de datos.
- **Desarrollador 4 (Alba):** Scrum, documentación y soporte general.

Disponibilidad:

- Solo tenemos 2 semanas, lo que limita el desarrollo.
- Para agilizar, podemos dividir tareas dentro del sprint en ciclos más cortos de 2-3 días.

Dificultades humanas:

- Sin experiencia previa en apps de citas o redes sociales, necesitaremos aprender sobre lógica de matchmaking.
- La sincronización entre backend y frontend podría ser un reto si no nos organizamos bien.

Conclusión humana: Viable, pero requerirá buena organización y distribución de tareas para evitar bloqueos.

4. Viabilidad Técnica

Lenguajes y herramientas:

- **Frontend:** React Native (Expo Snack para facilitar el despliegue rápido).
- **Backend:** Springboot y AWS (para manejar la lógica del servidor y hosting).
- **Base de datos:** MongoDB.
- **Autenticación:** AWS.
- **Matchmaking:** Algoritmo simple basado en coincidencias de tecnologías y experiencia.
- **Mockups y diseño:** Figma.

Pantallas principales:

1. **Pantalla de inicio:** Login, registro y recuperación de contraseña.
2. **Pantalla de swipe:** Donde los usuarios pueden deslizar perfiles a la derecha si les gusta o a la izquierda si no.
3. **Pantalla de perfil:** Para modificar información personal y preferencias.



4. **Pantalla de chats:** Para conversar con otros usuarios después de hacer match.
5. **Pantalla de modo premium:** Sección exclusiva para usuarios profesionales (a implementar en futuras versiones).
6. **Opciones de personalización:** Permite cambiar entre modo claro y oscuro, y elegir el idioma entre inglés y español.

Funcionalidades estrella:

- **Swipe:** Los usuarios pueden deslizar hacia la derecha si les gusta un perfil o hacia la izquierda si no.
- **Chats:** Una vez que dos usuarios hacen match, pueden iniciar una conversación dentro de la app.
- **Modo claro/oscuro:** Opción para cambiar la apariencia de la app.
- **Cambio de idioma:** Opción para elegir entre inglés y español.

Desafíos técnicos:

- Implementar correctamente la integración entre React Native y AWS.
- Implementar el swipe de forma fluida requiere usar librerías como React Native Gesture Handler.
- Manejar bien las conexiones y la lógica de emparejamiento en poco tiempo.
- Implementar la configuración de idioma y modo oscuro sin afectar la experiencia del usuario.

Futuras mejoras:

- **Funcionalidad premium para profesionales:** En futuras versiones, queremos incluir una opción premium con características avanzadas para usuarios profesionales.

Conclusión técnica: Viable si simplificamos la funcionalidad y nos enfocamos en lo esencial.



5. Conclusión General

"Love 4 Code" es un proyecto viable dentro del tiempo y recursos disponibles, siempre que:

- Seamos estratégicos con la división de tareas.
- Nos enfoquemos en lo esencial: perfiles, autenticación, swipe y chats.
- Optemos por herramientas que agilicen el desarrollo (React Native, AWS, Springboot, MongoDB).
- Evitemos funcionalidades avanzadas como pagos dentro de la app en esta fase inicial.



Plan de empresa

1. Introducción

1.1. Datos personales del promotor/es

- **Nombre y Apellidos:** Alba Sierra Vives
- **D.N.I./N.I.F.:** 48709660F
- **Email:** alsivi01@alumnatflorida.es
- **Teléfono:** 667357962

- **Nombre y Apellidos:** Hugo Miquel Vallés
- **D.N.I./N.I.F.:** 53882853V
- **Email:** humiva@alumnatflorida.es
- **Teléfono:** 622923562

- **Nombre y Apellidos:** Adrián Pons Choví
- **D.N.I./N.I.F.:** 53879548R
- **Email:** adpoch@alumnatflorida.es
- **Teléfono:** 622230884

- **Nombre y Apellidos:** Gabriel Carmona Palop
- **D.N.I./N.I.F.:** 20459794Y
- **Email:** gacapa@alumnatflorida.es
- **Teléfono:** 644365817

1.2. Datos profesionales del promotor/es

Los promotores del proyecto Love 4 Code son profesionales en el ámbito del desarrollo de software con formación en **Desarrollo de Aplicaciones Multiplataforma (DAM)**. Aunque su experiencia laboral es principalmente a nivel junior, cuentan con una sólida base teórica y práctica en tecnologías modernas de programación y desarrollo web y móvil. Además, su conocimiento del ecosistema digital y las necesidades del sector IT les permite desarrollar una propuesta de valor innovadora.



2. Definición del par producto/mercado

Producto/Servicio:

Love 4 Code es una aplicación que integra funcionalidades de una red social de citas con una plataforma profesional enfocada en programadores. Su objetivo principal es conectar a profesionales de la programación para generar oportunidades laborales, fomentar colaboraciones y facilitar conexiones personales dentro del sector IT.

Entre sus principales características destacan:

- Creación de perfiles con habilidades, tecnologías dominadas y experiencia.
- Algoritmo de emparejamiento basado en intereses y necesidades profesionales.
- Sistema de mensajería y foros especializados.
- Sección de ofertas de empleo y proyectos freelance.

Mercado objetivo:

Segmentos principales:

- Programadores junior que buscan empleo o colaboraciones.
- Empresas tecnológicas en búsqueda de talento IT.
- Reclutadores especializados en el sector tecnológico.
- Estudiantes de programación interesados en networking y aprendizaje colaborativo.

Ubicación: El lanzamiento inicial estará enfocado en **España**, con proyección de expansión a mercados internacionales conforme crezca la adopción de la plataforma.

3. Estudio de mercado

3.1. Análisis externo

Factores clave:

- **Tecnológicos:** Digitalización creciente del networking y empleo en tecnología.
- **Socioculturales:** Aumento del teletrabajo y comunidades virtuales de programadores.
- **Jurídicos:** Regulaciones de protección de datos (GDPR) y legislación sobre plataformas digitales.

3.2. Análisis interno

Fortalezas:

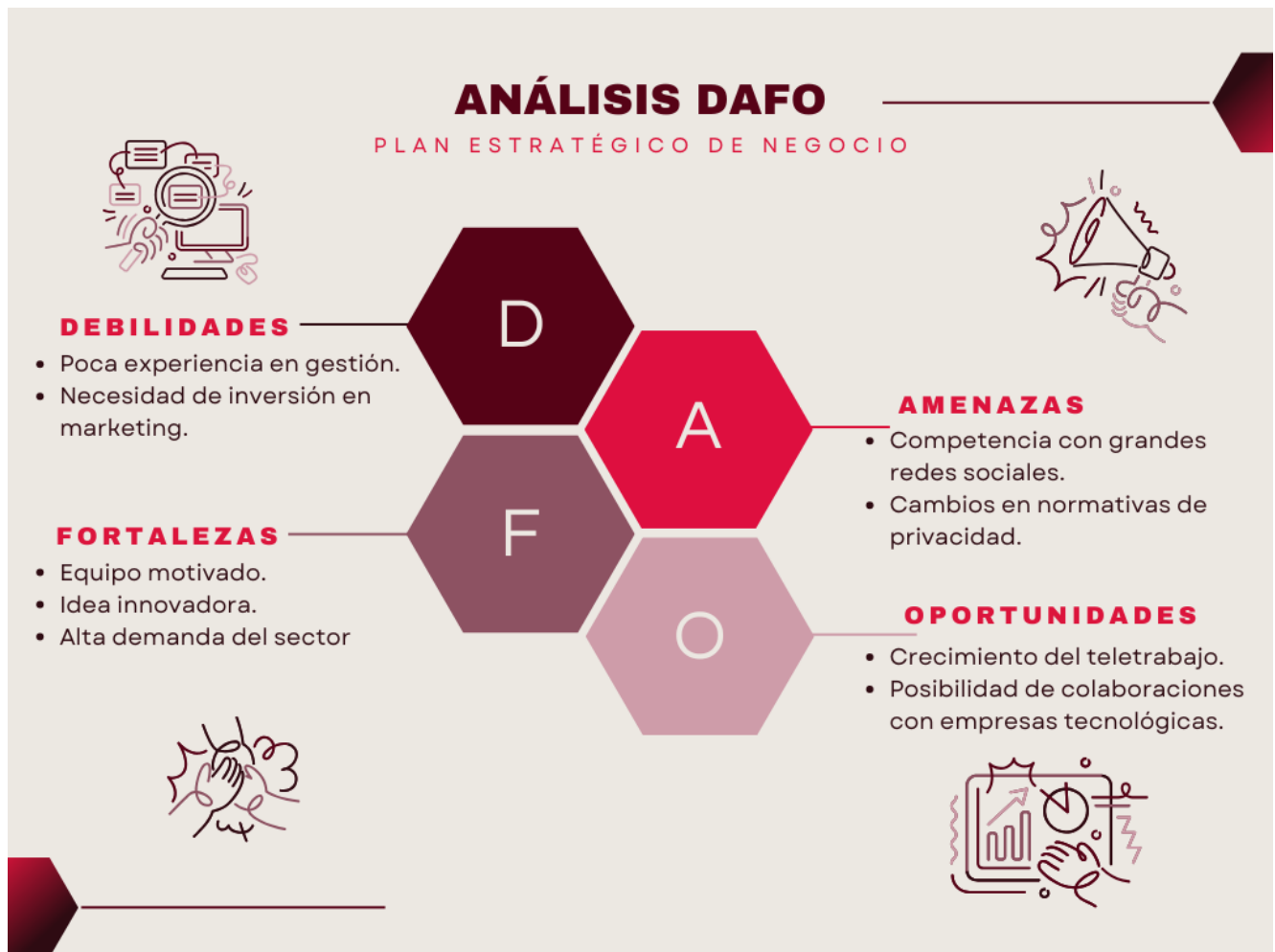
- Equipo con formación en tecnología y desarrollo de software.
- Propuesta de valor diferenciadora en el mercado.

- Modelo de monetización versátil (freemium, publicidad, comisiones).

Debilidades:

- Dependencia inicial de estrategias de marketing digital.
- Necesidad de captar una masa crítica de usuarios para generar interacciones valiosas.

3.3. Análisis DAFO





4. Plan de marketing

4.1. Producto/Servicio

- Aplicación web y móvil con funcionalidades avanzadas.
- Algoritmo de emparejamiento basado en tecnologías y preferencias.
- Espacios de networking, foros y proyectos colaborativos.

4.2. Precio

- **Modelo Freemium:** Acceso gratuito con funciones básicas.
- **Suscripción Premium:** Acceso a funciones avanzadas.
- **Publicidad y comisiones** por contrataciones en la plataforma.

4.3. Distribución

- Descarga a través de App Store, Google Play y web.
- Promoción en comunidades de programación (GitHub, Stack Overflow, Reddit).

4.4. Comunicación

- Redes sociales (LinkedIn, X, Instagram, TikTok).
- Marketing de contenido en blogs especializados.
- Influencers tech y embajadores de marca.

5. Forma jurídica y trámites

5.1. Elección de la forma jurídica

- **Forma jurídica:** Sociedad Limitada (SL).
- **Razón social:** Love 4 Code SL.
- **NIF:** (Por asignar).
- **Domicilio social:** Catarroja, España.
- **Capital social:** 3.000€.
- **Administradores:** Los cuatro socios fundadores.

5.2. Trámites de constitución y puesta en marcha

- Registro en el Registro Mercantil.
- Obtención de NIF y alta en Hacienda.
- Inscripción en la Seguridad Social.
- Registro de marca y protección de datos (GDPR).



6. Plan de operaciones

- Desarrollo y mantenimiento de la aplicación.
- Moderación de comunidad y seguridad de datos.
- Marketing y captación de usuarios.

7. Estructura de costes

- Desarrollo y hosting de la plataforma.
- Publicidad y marketing digital.
- Gestión de comunidad y atención al cliente.
- Costes legales y administrativos.



Términos y Condiciones de Love 4 Code

1. Objeto

Love 4 Code es una aplicación diseñada para conectar programadores con oportunidades laborales, colaboraciones y relaciones personales. La plataforma permite la creación de perfiles, interacción entre usuarios y acceso a contenido de networking.

2. Registro y Uso de la Plataforma

- Para utilizar nuestros servicios, debes registrarte proporcionando información verídica y mantener tu perfil actualizado.
- Eres responsable de la seguridad de tu cuenta y de no compartir tus credenciales de acceso.
- Nos reservamos el derecho de suspender o eliminar cuentas que incumplan nuestras normas.

3. Conducta del Usuario

Los usuarios deben:

- Usar la plataforma de manera respetuosa y profesional.
- No publicar contenido ofensivo, difamatorio o ilegal.
- No realizar actividades fraudulentas o dañinas dentro de la aplicación.

4. Propiedad Intelectual

Todo el contenido de Love 4 Code, incluyendo diseños, textos, imágenes y software, está protegido por derechos de propiedad intelectual y no puede ser utilizado sin autorización previa.



5. Limitación de Responsabilidad

Love 4 Code no se hace responsable por:

- Contenido generado por los usuarios.
- Daños derivados del uso de la plataforma.
- Disponibilidad ininterrumpida de los servicios.

6. Modificaciones de los Términos

Nos reservamos el derecho de actualizar estos términos en cualquier momento. Los usuarios serán notificados de cambios relevantes.

7. Legislación Aplicable

Estos términos se rigen por la legislación española. Cualquier controversia será resuelta en los tribunales de Catarroja, España.



Política de Protección de Datos de Love 4 Code

1. Introducción

En Love 4 Code nos tomamos en serio la privacidad de nuestros usuarios y garantizamos la protección de sus datos personales conforme al Reglamento General de Protección de Datos (GDPR).

2. Datos Recopilados

Recopilamos los siguientes datos:

- **Datos de registro:** Nombre, email, contraseña.
- **Datos de perfil:** Tecnologías dominadas, experiencia laboral.
- **Datos de interacción:** Mensajes, conexiones, preferencias.

3. Finalidad del Tratamiento

Utilizamos los datos recopilados para:

- Proporcionar y mejorar nuestros servicios.
- Conectar a los usuarios con oportunidades laborales y de networking.
- Garantizar la seguridad y el cumplimiento de nuestros términos de uso.

4. Compartición de Datos

No compartimos datos personales con terceros, excepto en los siguientes casos:

- Cuando sea necesario para la prestación del servicio.
- Cuando lo exija la ley.
- Con consentimiento explícito del usuario.



5. Seguridad de los Datos

Implementamos medidas técnicas y organizativas para proteger los datos contra accesos no autorizados, pérdida o alteración.

6. Derechos del Usuario

Los usuarios pueden ejercer sus derechos de acceso, rectificación, eliminación y oposición enviando un correo a love4codeapp@gmail.com.

7. Modificaciones de la Política

Nos reservamos el derecho de actualizar esta política y notificaremos a los usuarios sobre cualquier cambio significativo.

8. Contacto

Para dudas sobre la protección de datos, puedes contactarnos en love4codeapp@gmail.com.



Daily Scrum Meetings

Análisis de requerimientos

Obligatorios:

- Login y registrarse
- Swipe
- Chats
- Perfil
- Adaptar la app a español
- Link de "olvidaste contraseña"(web php)
- Modo oscuro/claro
- Colores app: #FAF8F6(blanco), #0B0804 (negro), #C7C7C7 (gris), #DC143C (rojo claro)
- Tipografía: Tahoma (accesibilidad)

Opcionales:

- Premium (Profesores)
- Notificaciones

Daily meeting 1 (03/02/2025)

Objetivos diarios: Preparar herramientas para los demás días centrarnos en codificar la app

- **Hugo:** Pantalla de login, signIn, y iniciar el swiper
- **Gabriel:** Objetos y tablas creados en flexy además de mockup inicial
- **Adrian:** Pantalla de chat, loading, perfil, teachers
- **Alba:** Plan de empresa, organizacion de trello

Daily meeting 2 (04/02/2025)

Objetivos Diarios : Funcionamiento de la Pantalla de Login y Registro de BackEnd , Olvidar Contraseña ¿?. Mockup en Figma.

- **Hugo:** Modo oscuro en toda la app, modificando funcionalidades de las pantallas
- **Gabriel:** AWS, FlexyGo, empezar API
- **Adrian:** empezar API, AWS
- **Alba:** Mockup en Figma con todas las pantallas que tenemos hasta hoy (modo claro y oscuro) y estudio de viabilidad económica, humana y técnica



Daily meeting 3 (05/02/2025)

Objetivos diarios: Juntar frontend y backend (login y sign), comienzo presentación, aclarar dudas frontend con Paco.

- **Hugo:** frontend (soluciona problemas de funcionamiento de registro y swipe)
- **Gabriel:** conectar la api con AWS, BBDD subida a AWS
- **Adrián:** crear maquina AWS, conectar la api con AWS, BBDD subida a AWS
- **Alba:** presentación, actualización mockup, trello

Daily meeting 4 (06/02/2025)

Objetivos diarios: acabar el registro login y olvidar contraseña

- **Hugo:** frontend (soluciona problemas de funcionamiento)
- **Gabriel:** login y registro unificados el backend y frontend, preparación de edicion de perfil y swipe
- **Adrián:** login y registro unificados el backend y frontend, preparación de edicion de perfil y swipe
- **Alba:** Florida expo, folleto, modificar mockup

Daily meeting 5 (07/02/2025)

Objetivos diarios:

- Tokens: realizar que no se puedan realizar peticiones POST, PUT, DELETE, sin haber token. Comprobar que funciona el token haciendo una petición post desde navegador (que no tiene token).
- Slider: array cartas con la info de la API
- Editar perfil (petición post) para actualizar los datos del usuario en la BD
- Olvidar contraseña: realizar la actualización de la contraseña para después en la edición del perfil añadir un botón que sea cambiar contraseña.
- Revisar todo lo necesario para florida expo
- Databasename al crearse: usuario ya existente, ponga otro.
- Crear qr con RRSS provisionales para florida expo.
- **Hugo:** RRSS, edición perfiles, sincronizar datos, mostrar perfil, cerrar sesión.
- **Gabriel:** recuperacion psw, comprobar que no se repitan usuarios, perfil funcionando, edición de perfil
- **Adrián:** recuperacion psw, comprobar que no se repitan usuarios, perfil funcionando, edición de perfil
- **Alba:** Modificar plan de empresa, diseño Florida Expo, RRSS



Daily meeting 6 (10/02/2025)

Objetivos diarios:

- Acabar dislike
- Decorar alerta de match
- Acabar logica de cartas (que no salgan usuarios que has rechazado/aceptado)
- Listar chats (usuarios con match)
- Realizar un mensaje predefinido al compartir la app
- Hacer folleto y cartel florida expo
- Crear qr.
- **Hugo:** pantalla de chat, pantalla terminos y condiciones y arreglar fallos de funcionamiento frontend
- **Gabriel:** mensaje matches, like y dislike, creacion de endpoints, pantalla mensaje compartir app, logica carta repetidas
- **Adrián:** arreglar bug imagenes, like y dislike, creación endpoints, mensaje matches, logica carta repetidas
- **Alba:** Modificar plan de empresa, cartel y folleto florida expo, ampliar linktree, documento de términos, condiciones y protección de datos

Daily meeting 7 (11/02/2025)

Objetivos diarios:

- Realizar funcionamiento de los chats
- Lista de chat segun el usuario
- Chat en tiempo real (string)
- Imágenes y audios
- Funcionalidades premium de chats de profesores
- Ampliar mockup
- HCM con FlexyGo
- Retoques: cambiar contraseña, cerrar sesión, guardar perfil editado
- **Hugo:** crear pantalla de premium, cambiar contraseña y arreglar funcionalidades
- **Gabriel:** lista de chats, envio de chats en tiempo real, imagenes, nombres
- **Adrián:** lista de chats, envio de chats en tiempo real, imagenes, nombres
- **Alba:** Ampliar mockup con nuevas funcionalidades y pantallas actualizadas, hacer HCM con FlexyGo



Daily meeting 8 (12/02/2025)

Objetivos diarios:

- Tokens
- Traducción
- Logout
- Cargar cartas al volver a pantalla de swipe
- Cargar último mensaje del chat al entrar en la pantalla
- Cargar chats nuevos al cargar la pantalla
- Ordenar por fecha de mensaje los chats
- Folleto para presentación: español e inglés
- Hacer Readme
- Botones like y dislike
- premium
- **Hugo:** Traducción de la app, corregir errores de funcionamiento, modificar diseño front.
- **Gabriel:** reestructurar la máquina de AWS, chats en tiempo real, mejora de la API, swipe en tiempo real, detalles de interfaz, logout y tokens
- **Adrián:** reestructurar la máquina de AWS, chats en tiempo real, mejora de la API, swipe en tiempo real, detalles de interfaz, logout y tokens
- **Alba:** Folleto para presentación en inglés y en español, documento Readme Github, encargar camisetas serigrafiadas, comprar material para el elevator pitch

Daily meeting 9 (13/02/2025)

Objetivos diarios:

- Deshabilitar botones swipe
- Tutorial uso app
- Arreglar diseño chats
- Traducción en toda la app
- Completar hcm en flexygo
- Organizar tareas Trello
- Documentación: HCM
- Premium
- **Hugo:** Deshabilitar botones swipe, tutoriales, traducción
- **Gabriel:** Arreglar diseño chats, funcionamiento lista profesores, premium
- **Adrián:** Arreglar diseño chats, funcionamiento lista profesores, premium
- **Alba:** completar HCM en FlexyGo, documentación: HCM, organizar Trello, recoger camisetas



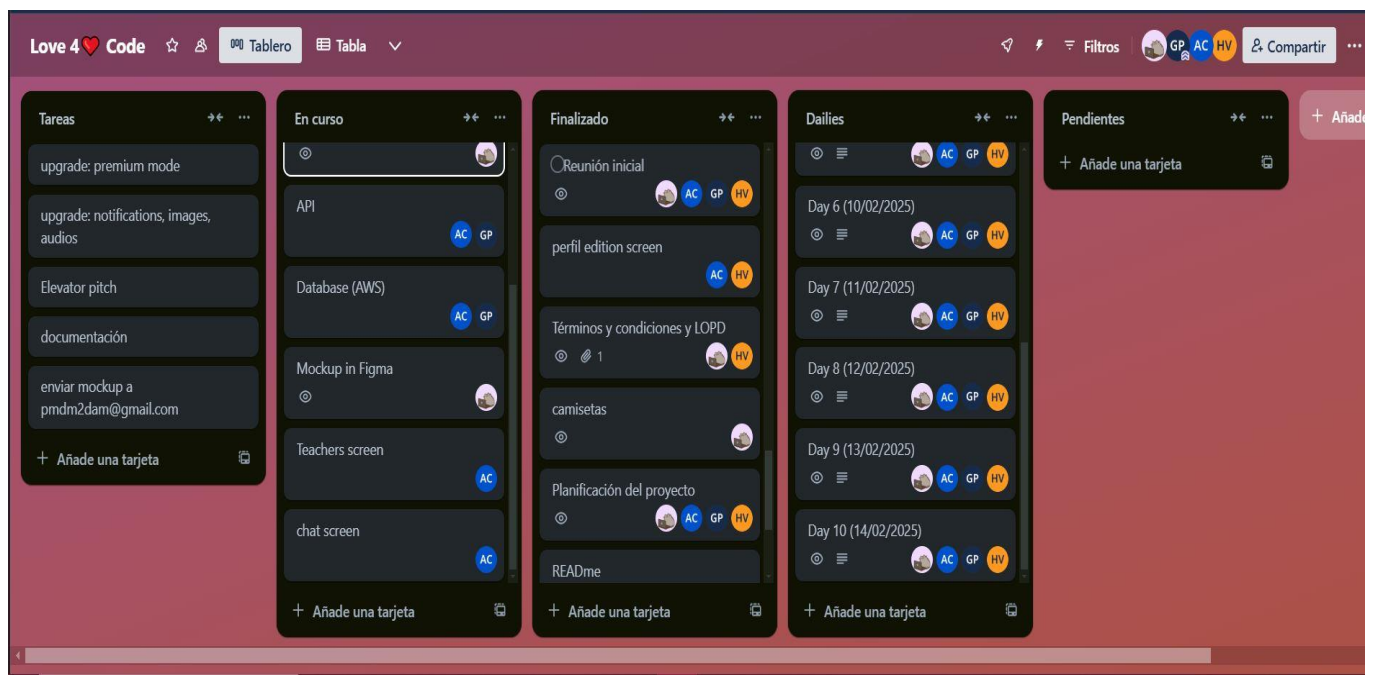
Daily meeting 10 (14/02/2025)

Objetivos diarios:

- Traducción (Estructurar todo)
- APK
- Teclado chats
- Proteger BBDD
- Mensaje de compartir
- Poner el linktree
- Github

- **Hugo:** Traducción, teclado chats, mensaje de compartir
- **Gabriel:** APK, proteger bbdd, Github
- **Adrián:** APK, proteger bbdd, Github
- **Alba:** terminar fichero README, terminar de trabajar Trello, Github

Trello: <https://trello.com/b/cbzXnrTN>



Documentación BBDD:

Base de Datos:

Introducción:

La base de datos utilizada en este proyecto está implementada en MongoDB, una base de datos NoSQL orientada a documentos. Su estructura flexible nos permite almacenar datos en formato JSON/BSON, facilitando la escalabilidad y eficiencia del sistema.



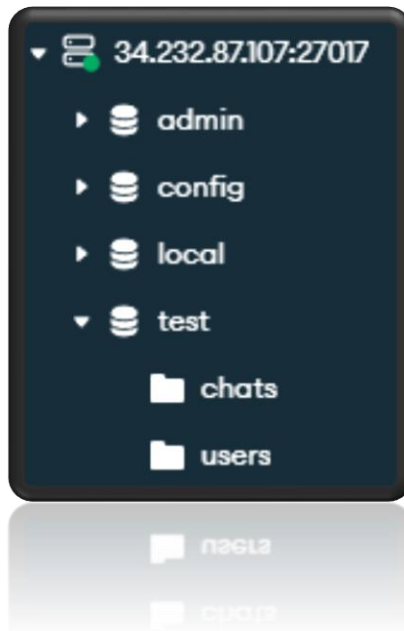
mongoDB



spring
BOOT

Esta base de datos NoSQL esta formada por dos colecciones:

- Users : Almacenar Información de los Usuarios de la APP Love4Code.
- Chats : Guardaremos los respectivos Chats para que en todo momento la conversación se quede guardada entre el usuario1 y usuario2.



Estructura de la BBDD:

Usuarios (users)

Esta colección almacena la información de los usuarios registrados en el sistema.

- **_id** (ObjectId): Identificador único del usuario.
- **name** (String): Nombre del usuario.
- **database_name** (String): Nombre interno de la base de datos del usuario.
- **gender** (String): Género del usuario.
- **age** (Number): Edad del usuario.
- **programming_languages** (String): Lenguajes de programación que conoce.
- **premium** (Boolean): Indica si el usuario tiene cuenta premium.
- **teacher** (Boolean): Indica si el usuario es profesor.
- **password** (String): Hash de la contraseña del usuario.
- **email** (String): Correo electrónico del usuario.
- **matched** (Array): Lista de usuarios con los que ha coincidido.
- **liked** (Array): Lista de usuarios que ha marcado como "me gusta".
- **disliked** (Array): Lista de usuarios que ha marcado como "no me gusta".
- **image** (String): Imagen de perfil en base64.
- **premiumExpirationDate** (Date): Fecha de expiración de la suscripción premium.
- **token** (String): Token de autenticación del usuario.

```
_id: ObjectId('67af61ab555c9f1fa71fd846')
name: "Adrian Pons"
database_name: "pons"
gender: "Male"
age: 19
programming_languages: "JavaScript,Python,Java"
premium: true
teacher: true
password: "03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4"
email: "adrian@gmail.com"
matched: Array (1)
liked: Array (4)
disliked: Array (1)
image: "data:image/jpeg;base64,/9j/4QBYRXhpZgAATU0AKgAAAQABAEEAAQAAAAABAAADhAE..."
premiumExpirationDate: "2025-03-14"
token: "269a67e1-24eb-41d7-9232-7414f01e759b"
_class: "main.model.User"
```



Chats (chats)

Esta colección almacena los mensajes intercambiados entre los usuarios.

- **_id** (ObjectId): Identificador único del chat.
- **userId1** (ObjectId): ID del primer usuario participante.
- **userId2** (ObjectId): ID del segundo usuario participante.
- **messages** (Array): Lista de mensajes enviados en el chat.
 - **message** (String): Contenido del mensaje.
 - **timestamp** (Date): Fecha y hora en que se envió el mensaje.

```
  _id: ObjectId('67af6816555c9f1fa71fd849')
  userId: "67af64d8555c9f1fa71fd847"
  userId2: "67af61ab555c9f1fa71fd846"
  messages: Array (19)
    0: Object
    1: Object
    2: Object
    3: Object
    4: Object
    5: Object
    6: Object
    7: Object
    8: Object
    9: Object
    10: Object
    11: Object
    12: Object
    13: Object
    14: Object
    15: Object
    16: Object
    17: Object
    18: Object
  _class: "main.model.Chat"
```

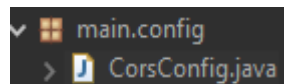
Documentación API:

Descripción General

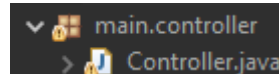
Esta API ha sido desarrollada utilizando **Spring Boot** y **MongoDB** como base de datos NoSQL. Está diseñada para gestionar usuarios, interacciones entre ellos y sistemas de chat en tiempo real.

La API se organiza en **cinco paquetes principales**, cada uno con una responsabilidad específica:

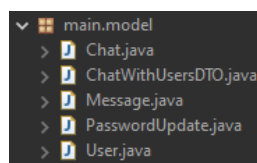
- **Config** : Package que contiene el fichero CorsConfig.java el el cual configura **CORS (Cross-Origin Resource Sharing)** para permitir solicitudes desde cualquier dispositivo o frontend.



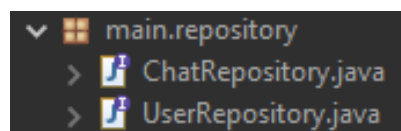
- **Controller**: Este paquete contiene todas las **clases controladoras** que definen los **endpoints** de la API. Los controladores manejan las peticiones HTTP (GET, POST, PUT, DELETE) y se comunican con los servicios para procesar la lógica de negocio



- **Model**: Define las clases que representan documentos de MongoDB para poder ser utilizados en el Controller.



- **Repository**: Repositorios los cuales añadiremos queries por defecto las cuales podremos utilizar de primera mano o tambien podemos realizar nuestras propias queries personalizables las cuales ayudan mucho a encontrar o extraer justamente los datos que queramos. En este caso tenemos dos : (UserRepository y ChatRepository).





- Cadena de Conexión en la API "backend":

```
application.properties ×  
1 spring.data.mongodb.host= localhost  
2 spring.data.mongodb.port= 27017  
3 spring.data.mongodb.database= love4code
```

ENDPOINTS:

Gestión de Usuarios

1. Actualizar a usuario premium (PUT API/user/update-premium)

- Requiere un token de autenticación válido.
- Actualiza la suscripción premium de un usuario, extendiéndola según el plan elegido (1 mes, 3 meses o 1 año).
- Si el usuario ya es premium, la fecha de expiración se extiende desde su fecha actual de vencimiento.
- Devuelve los detalles actualizados del usuario con su estado premium.

2. Obtener profesores (GET API/users/teachers)

- Requiere autenticación.
- Devuelve una lista de usuarios marcados como profesores.
- Incluye detalles como nombre, edad, género, lenguajes de programación, estado premium y su imagen.

3. Obtener detalles de un usuario (GET API/user/{userId})

- Requiere autenticación.
- Busca un usuario por su ID y devuelve su información.
- Incluye detalles como nombre, género, edad, lenguajes de programación y estado premium.

4. Obtener todos los usuarios (GET API/users)

- Requiere autenticación.
- Devuelve una lista de usuarios disponibles para interactuar, excluyendo aquellos ya marcados como "like", "dislike" o "match".



5. Actualizar información de usuario (PUT API/update-user)

- Requiere autenticación.
- Permite a un usuario modificar sus datos personales como nombre, edad, género, imagen, etc.

6. Registro de usuario (POST API/register)

- Permite crear un nuevo usuario en la base de datos.
- Verifica si el usuario ya existe antes de registrarlo.

7. Inicio de sesión (POST API/login)

- Verifica las credenciales de un usuario.
- Si son correctas, genera un token de sesión y lo devuelve junto con los detalles del usuario.

8. Cerrar sesión (POST API/logout)

- Requiere autenticación.
- Elimina el token de sesión del usuario, cerrando su sesión.

9. Verificar si un correo está registrado (POST API/verify-email)

- Comprueba si un email ya está en uso en la base de datos.
- Devuelve un mensaje de error si no se encuentra.

10. Restablecer contraseña (PUT API/reset-password)

- Busca un usuario por su email y actualiza su contraseña en la base de datos.

Gestión de Chats

11. Obtener los chats de un usuario (GET API/chat/user-chats)

- Requiere autenticación.
- Devuelve todos los chats en los que participa el usuario.

12. Crear un chat (POST API/chat/create)

- Crea un chat entre dos usuarios si no existe uno previo.
- Devuelve un error si el chat ya está registrado.



13. Enviar un mensaje (POST API/chat/send-message)

- Requiere autenticación.
- Agrega un mensaje a un chat existente.
- Devuelve un error si el chat no existe.

14. Obtener los mensajes de un chat (GET API/chat/messages)

- Requiere autenticación.
- Devuelve todos los mensajes de un chat específico.

Interacciones entre Usuarios

15. Dar like a un usuario (PUT API/like)

- Requiere autenticación.
- Marca a otro usuario con un "like".
- Si ambos usuarios se han dado "like", se crea automáticamente un chat entre ellos.

16. Dar dislike a un usuario (PUT API/dislike)

- Requiere autenticación.
- Marca a un usuario con "dislike" para que no vuelva a aparecer en las recomendaciones.



Tecnologías utilizadas:

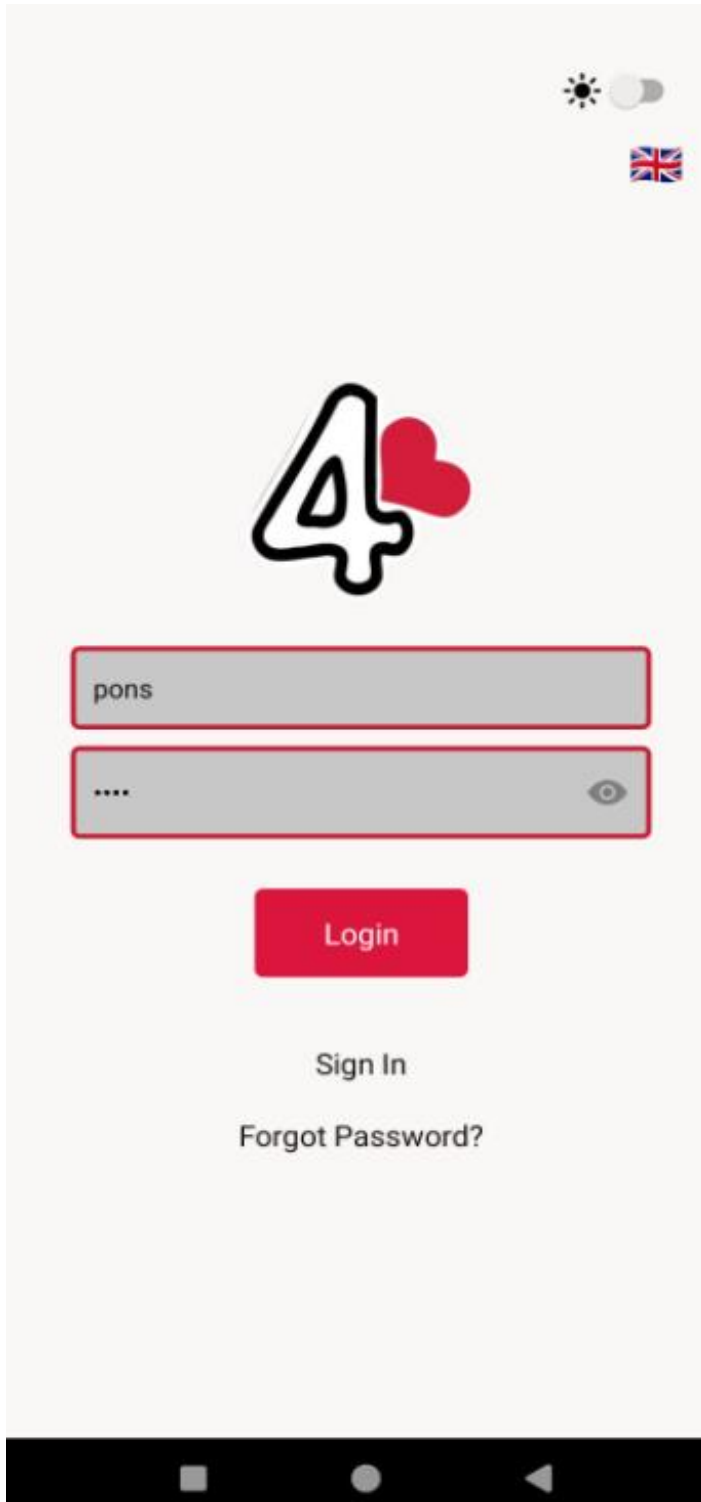
- **Spring Boot** → Para la estructura del backend.
 - **Spring Data MongoDB** → Para la interacción con la base de datos.
 - **MongoDB** → Base de datos NoSQL utilizada.
 - **CORS** → Para permitir peticiones desde diferentes dispositivos.
-

Flujo de Funcionamiento de la API:

1. **Un usuario se registra** en la API con su email y contraseña.
2. **El usuario inicia sesión**, y la API le devuelve un **token JWT**.
3. **Con el token**, el usuario puede acceder a los endpoints protegidos.
4. **Los usuarios pueden interactuar** (dando "like" o "dislike" a otros).
5. **Si dos usuarios coinciden en "like"**, se crea un **chat automáticamente**.
6. **Los usuarios pueden enviar y recibir mensajes** en tiempo real.

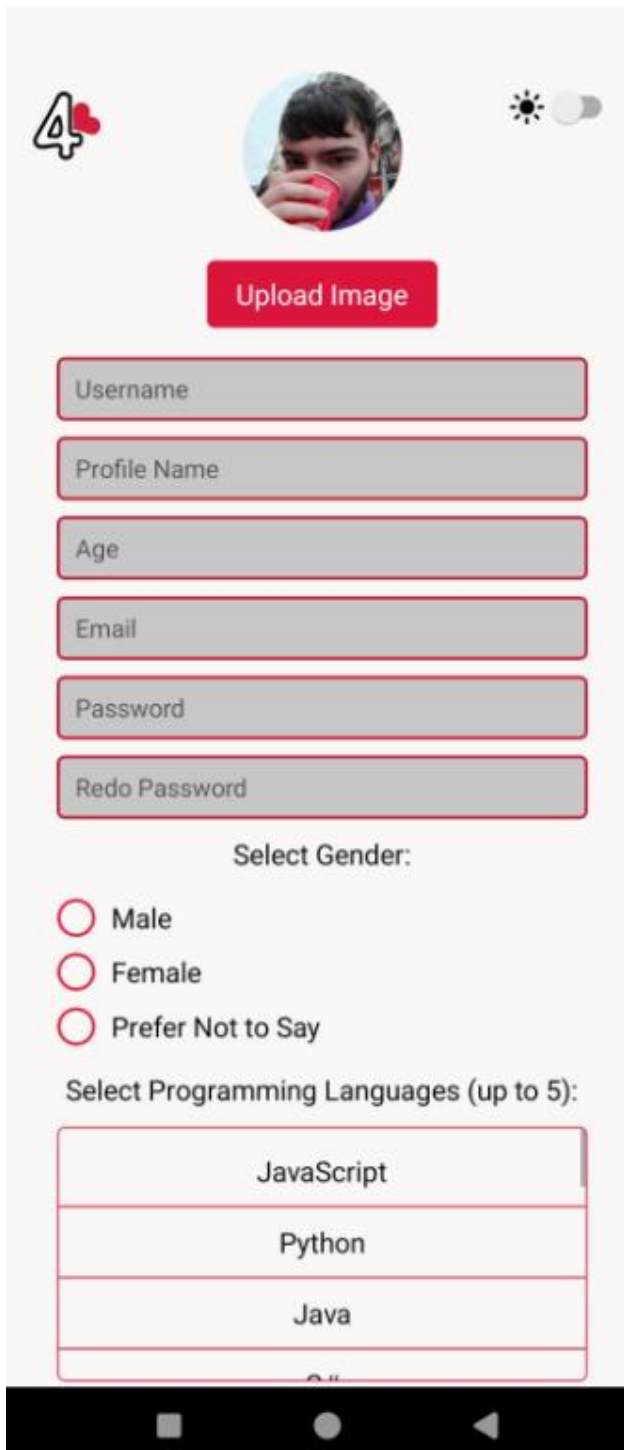
Pantallas App

Login



Esta pantalla es la del “login”, esta pantalla es la primera que sale en nuestra aplicación, al ser la primera, tenemos disponible la funcionalidad de alternar tanto de modo oscuro como claro y de idioma inglés a español, ya dentro de las funcionalidades de la pantalla, tenemos dos cuadros de texto, en los cuales introduciremos tanto el usuario como la contraseña de nuestro usuario, además dispondremos de un link de registro y otro de recuperar la contraseña en caso de que se nos haya olvidado

SignIn



4

Upload Image

Username

Profile Name

Age

Email

Password

Redo Password

Select Gender:

☐ Male

☐ Female

☐ Prefer Not to Say

Select Programming Languages (up to 5):

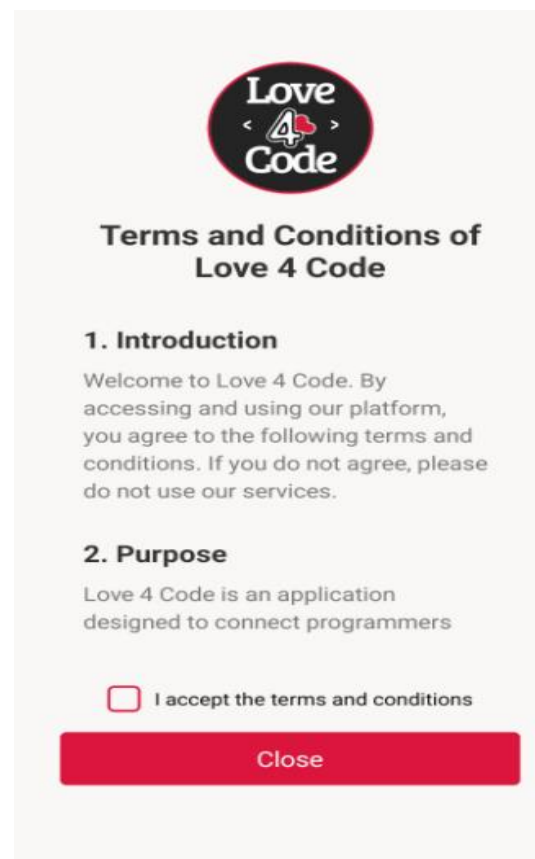
JavaScript

Python

Java

Esta es la pantalla de SignIn, en esta pantalla indicaremos nuestro nombre tanto para conectarse a la aplicación (username) como el nombre del perfil (profile name) además de nuestra edad, email, contraseña, genero y lenguajes de programación (solo podremos elegir 5) además, tendremos que aceptar los términos y condiciones de la aplicación

Al clicar el logo, te devolverá a la pantalla de login



Love 4 Code

Terms and Conditions of Love 4 Code

1. Introduction

Welcome to Love 4 Code. By accessing and using our platform, you agree to the following terms and conditions. If you do not agree, please do not use our services.

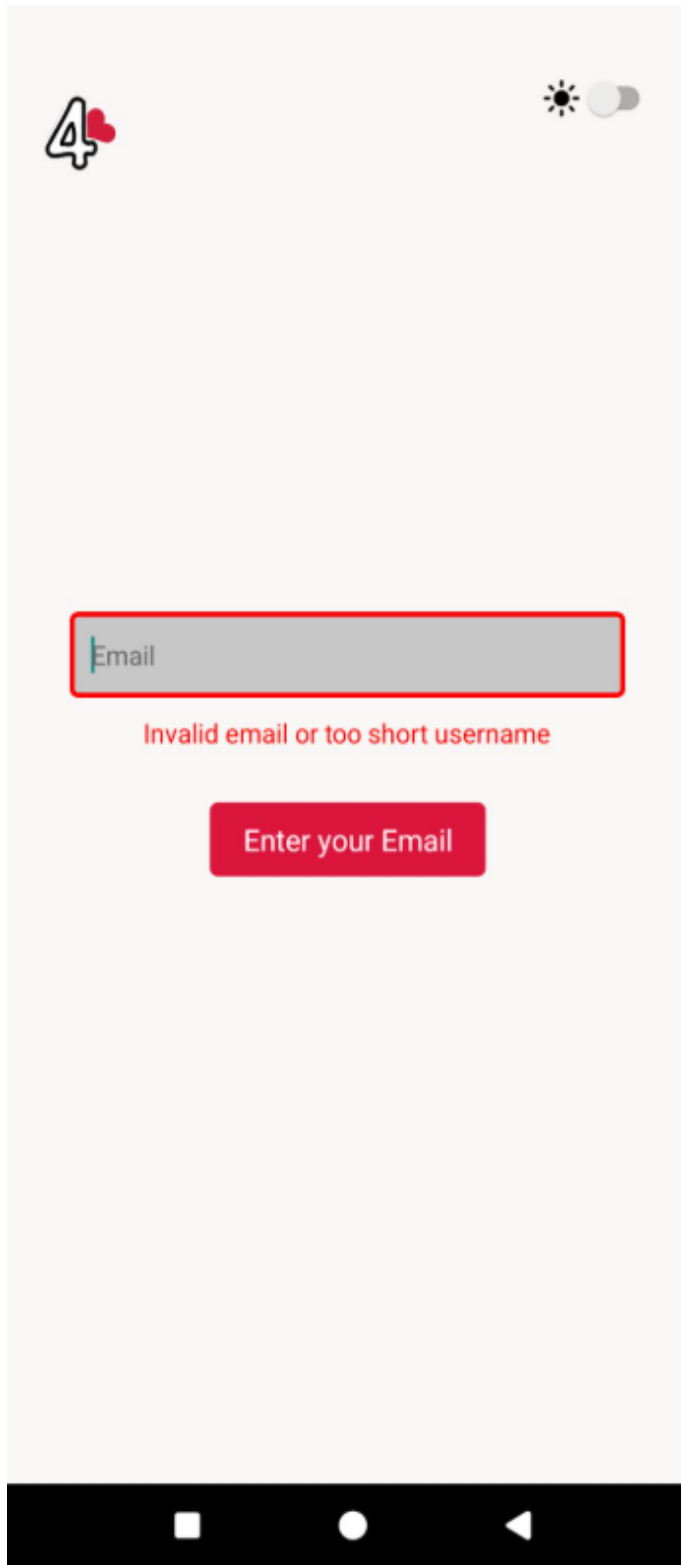
2. Purpose

Love 4 Code is an application designed to connect programmers

☐ I accept the terms and conditions

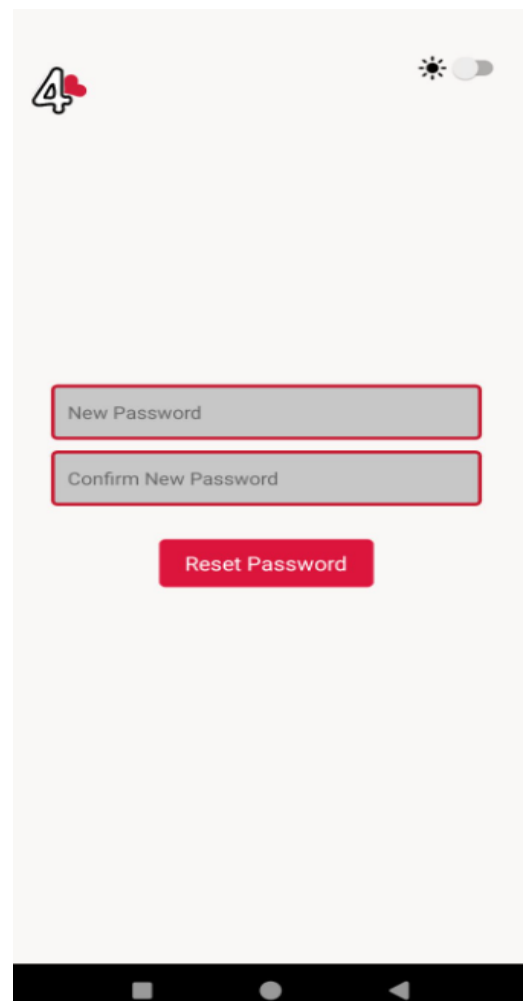
Close

Forgot password



The screenshot shows a mobile app interface for the 'Forgot password' screen. At the top left is a logo with the number '4' and a red heart. At the top right is a sun icon and a toggle switch. In the center, there is a text input field with the placeholder text 'Email'. Below the input field, a red error message reads 'Invalid email or too short username'. Below the error message is a red button with the text 'Enter your Email'. At the bottom of the screen is a black navigation bar with three white icons: a square, a circle, and a triangle.

En esta pantalla que se accede desde el login, encontraremos la opción de introducir nuestro email para poder restablecer nuestra contraseña, en caso de ser correcta te redireccionará a la pantalla de abajo donde podrás rescribir tu contraseña.



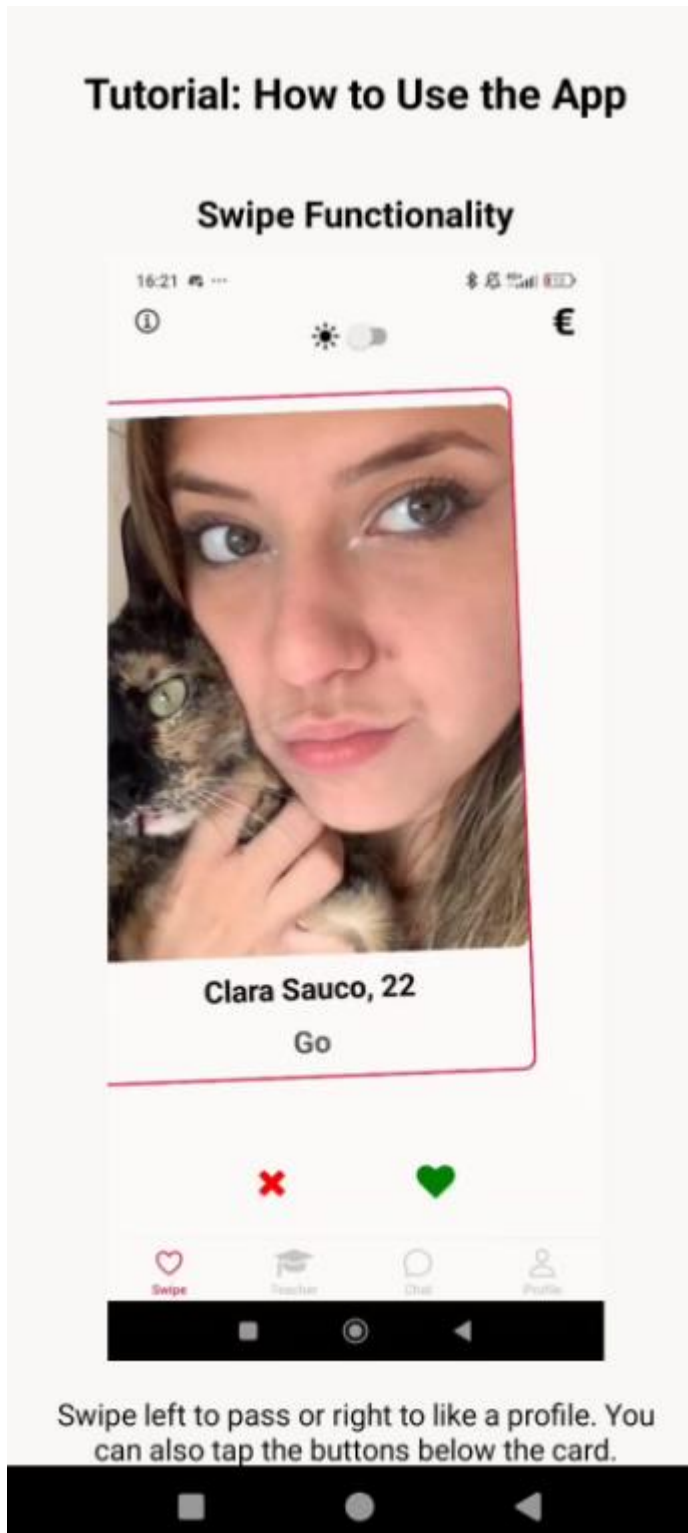
The screenshot shows the same mobile app interface for the 'Forgot password' screen. At the top left is the logo with the number '4' and a red heart. At the top right is the sun icon and the toggle switch. Below the email input field, there are two more text input fields: the first is labeled 'New Password' and the second is labeled 'Confirm New Password'. Below these two fields is a red button with the text 'Reset Password'. At the bottom of the screen is the same black navigation bar with three white icons.

Swipe



El swipe es la primera pantalla que se te muestra al entrar correctamente a nuestra aplicación, ofreciéndote tanto diversos botones como el tutorial, premium o cambio de tono del móvil, o la opción de deslizar tanto a la derecha para aceptar al usuario, si los dos usuarios se aceptan, se creará un “match” y se creará un chat conjunto en la sección de “chat”, como a la izquierda para rechazarlo, al rechazarlo no te volverá a mostrar el usuario que hayas rechazado.

Tutorial



Al darle al icono de la información en el swipe ⓘ se te despliega esta pantalla, la cual te muestra como funciona nuestra aplicación, te explica tanto como rechazar o aceptar a personas, como la creación de un chat debido a un match y la interacción con ellos,



Pago Premium

Choose Your Plan

1 Month Plan

4.99€

3 Months Plan

11.99€

1 Year Plan

49.99€

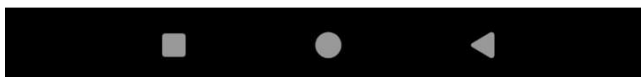
Go Back

En la pantalla de pago tenemos 4 botones:

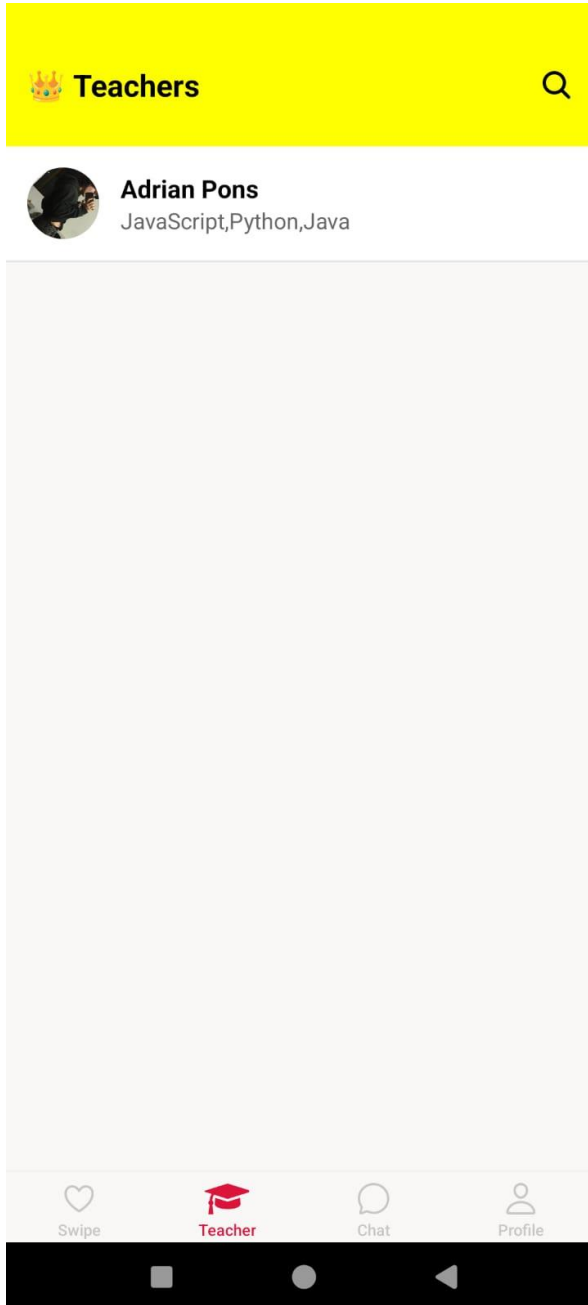
-1 Month Plan: Simula el pago de la suscripción premium de 1 mes y nos otorga el estatus de premium,

-3 Months Plan: Simula el pago de la suscripción premium de 3 meses y nos otorga el estatus de premium,

-1 Month Plan: Simula el pago de la suscripción premium de 1 mes y nos otorga el estatus de premium,



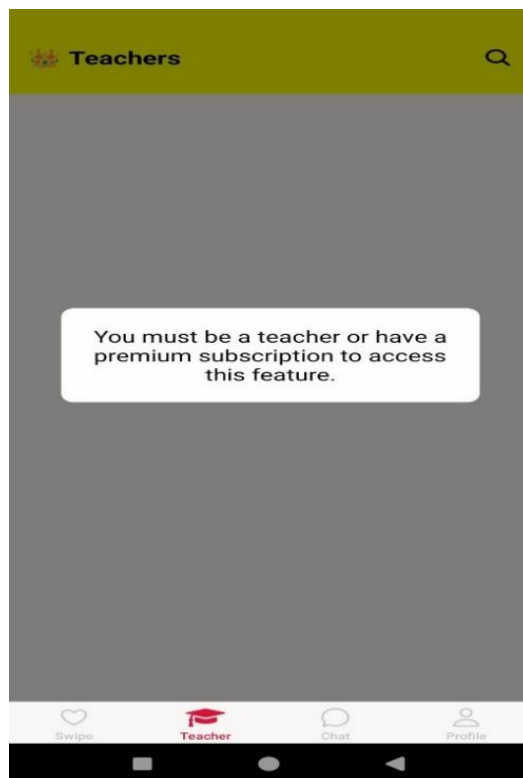
Profesores



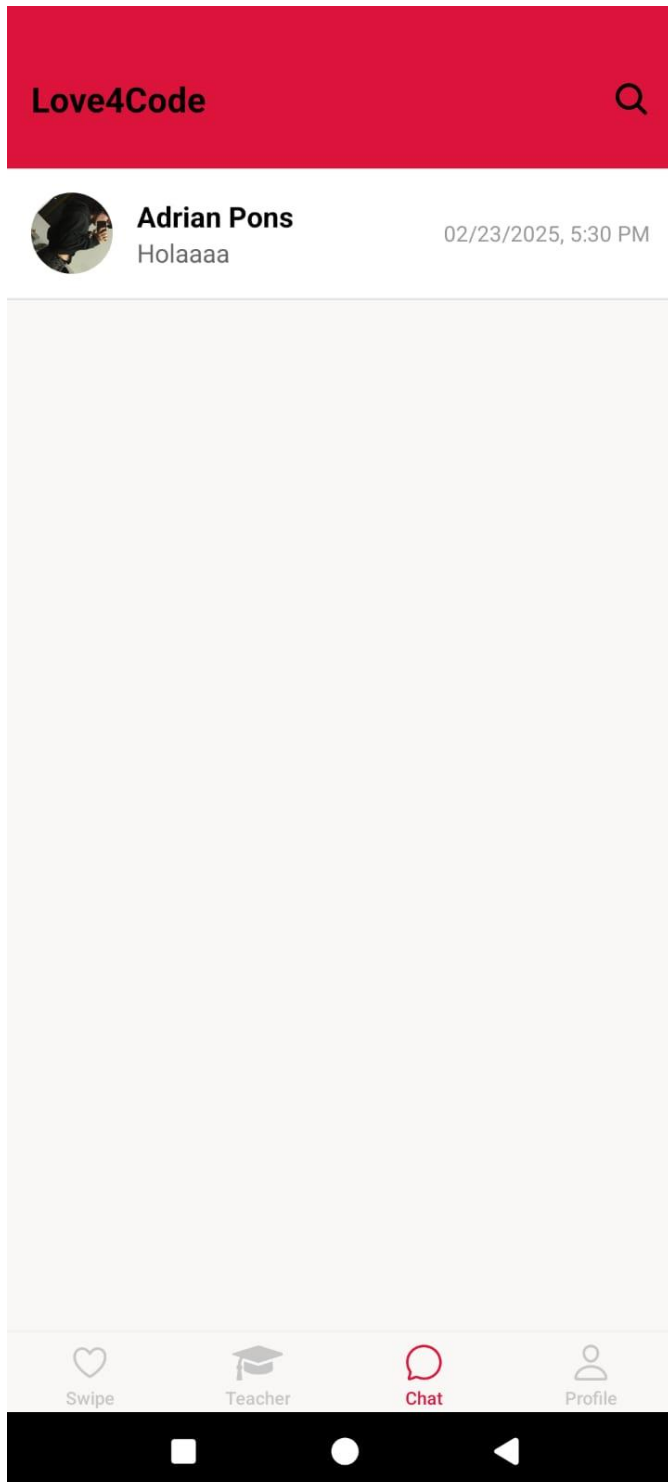
En la pantalla de profesores tenemos un listado de todos los profesores con los que cuenta nuestra aplicación.

Al clicar sobre un profesor, se añadirá su chat a la pantalla de chats de tu usuario.

Necesitas ser premium para ver esta pantalla



Listado Chats



En la pantalla de listado de chats tenemos un listado de todos los chats en los que cuenta tu usuario.

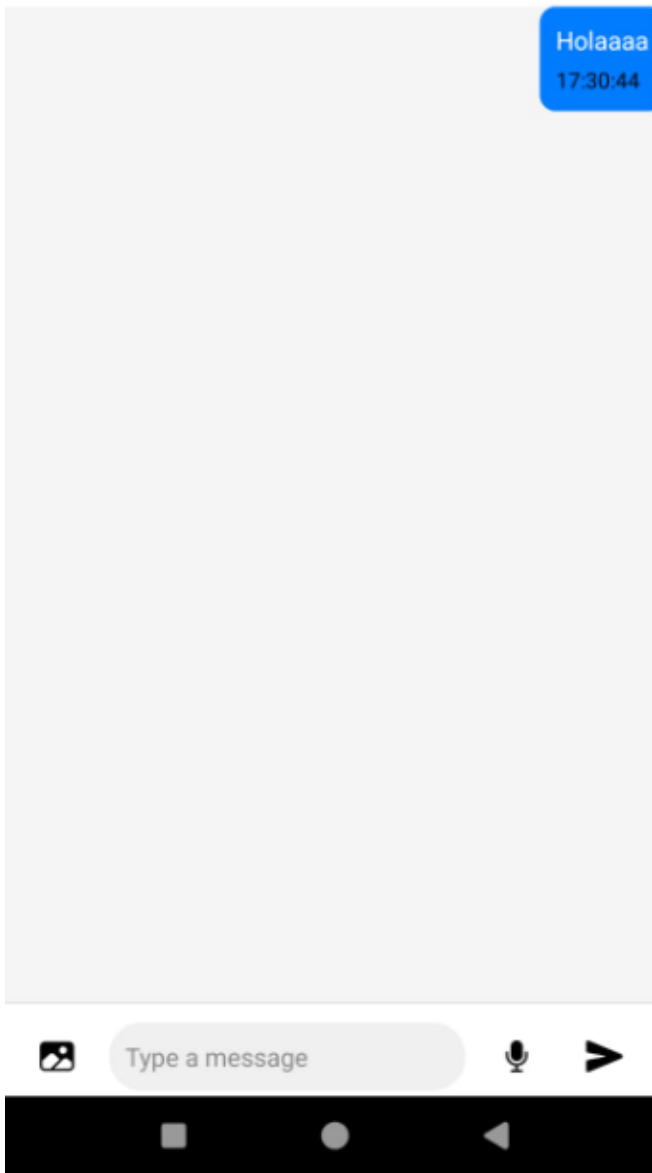
Cada chat muestra el último mensaje enviado, la foto y nombre del otro participante y la fecha del ultimo mensaje enviado.

Al clicar sobre un chat, se abre la pantalla de ese chat.

ChatScreen

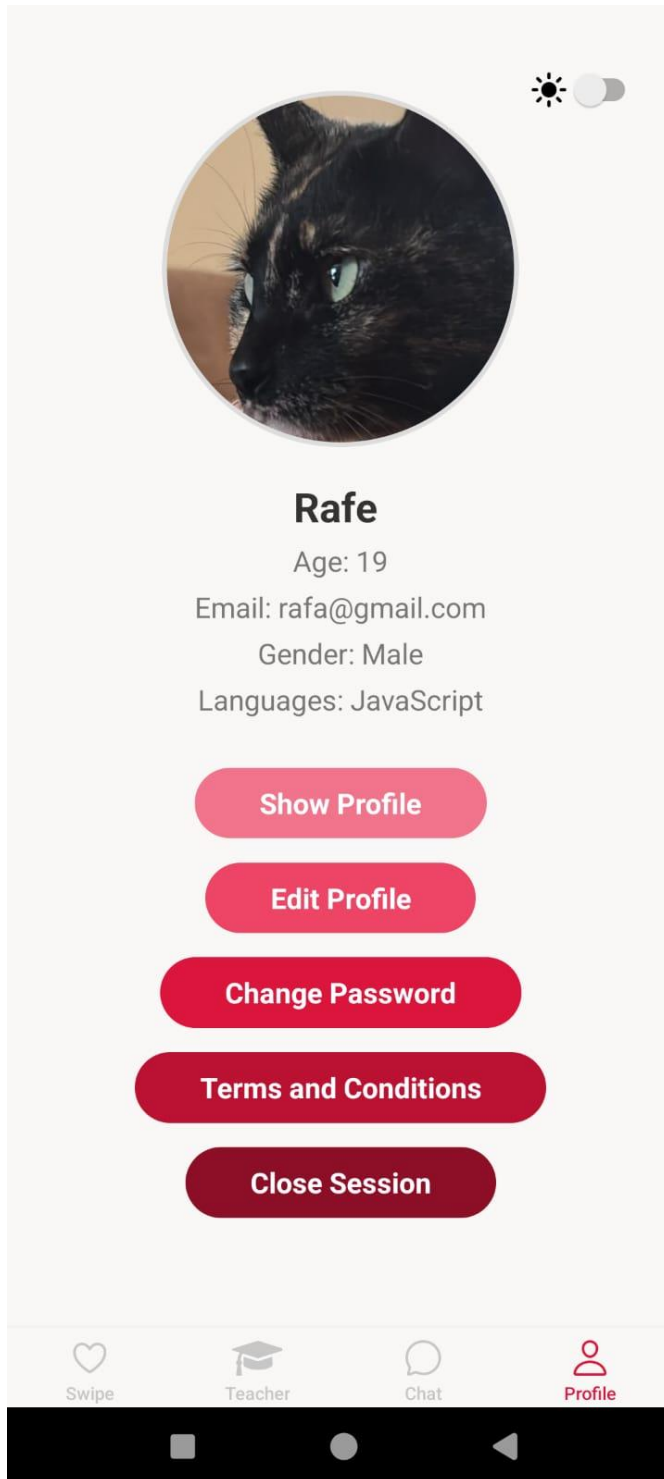


Adrian Pons



Al clicar en cualquiera de los chats, se nos abrirá esta pantalla, la cual nos mostrara tanto la imagen como el nombre del usuario con el cual nos estemos hablando, al introducir un mensaje, se nos mostrar tanto nuestro mensaje de diferente color y alineado a cada lado dependiendo si es receptor o emisor y la hora, para implementaciones futuras hemos decidido introducir como introducir imágenes o enviar audios por mensaje.

Perfil



En la pantalla de perfil tenemos la información de nuestro usuario y 5 botones:

- Show Profile: muestra nuestra carta de usuario.

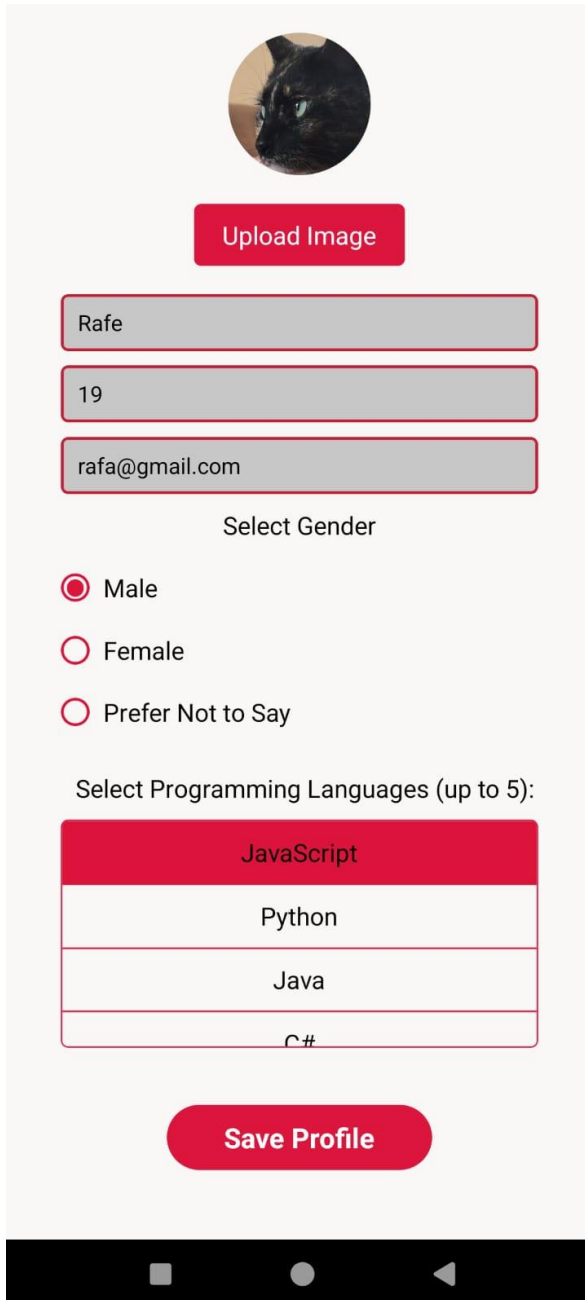
- Edit profile: abre la pantalla de edición de perfil.

- Change password: te lleva a la pantalla de cambiar contraseña.

- Terms and conditions: abre la pantalla de Términos y condiciones.

- Close Session: cierra la sesión.

Editar perfil



A mobile application screen for editing a profile. At the top is a circular profile picture of a black cat. Below it is a red button labeled 'Upload Image'. The form contains three text input fields: the first contains 'Rafe', the second contains '19', and the third contains 'rafa@gmail.com'. Below these is a 'Select Gender' section with three radio button options: 'Male' (selected), 'Female', and 'Prefer Not to Say'. This is followed by a 'Select Programming Languages (up to 5):' section with a list of languages: 'JavaScript' (highlighted in red), 'Python', 'Java', and 'C#'. At the bottom is a red button labeled 'Save Profile'. The screen has a black Android navigation bar at the very bottom.

Upload Image

Rafe

19

rafa@gmail.com

Select Gender

☒ Male

☐ Female

☐ Prefer Not to Say

Select Programming Languages (up to 5):

JavaScript

Python

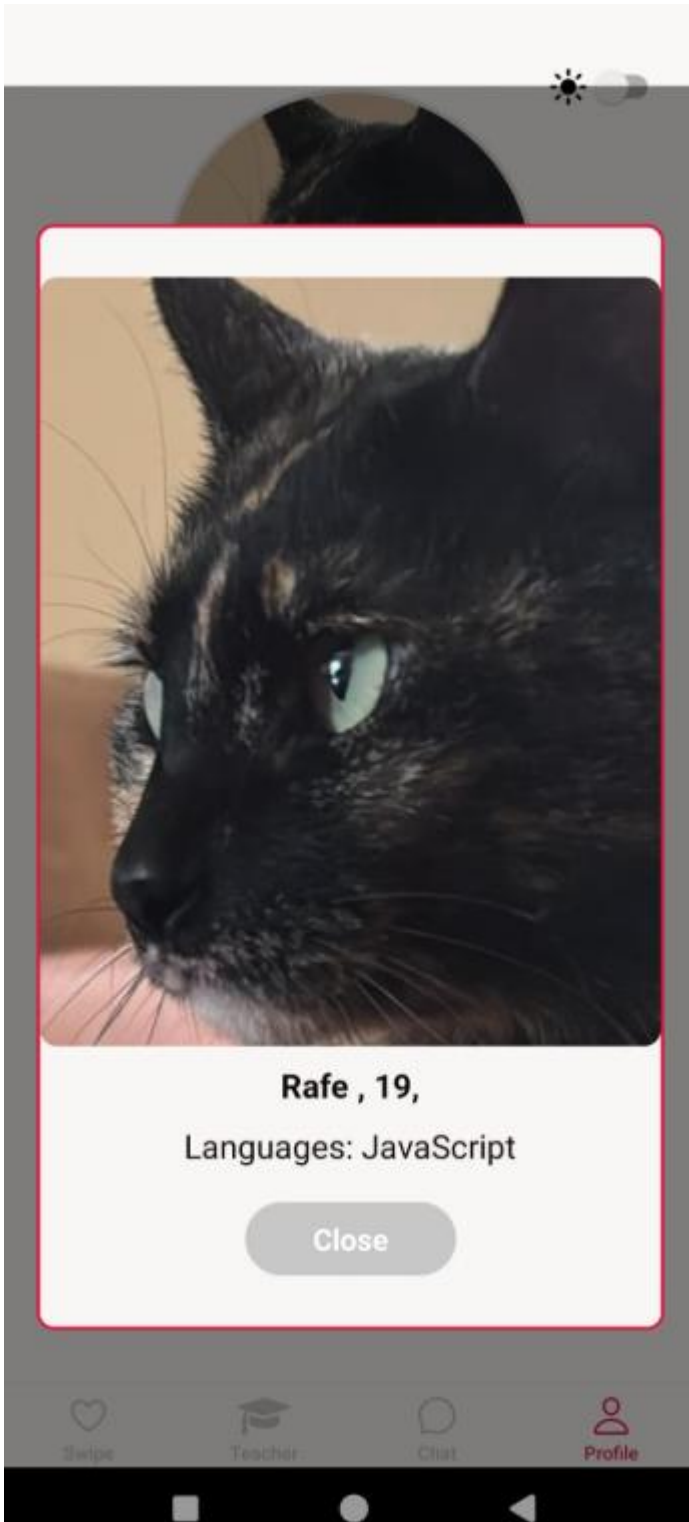
Java

C#

Save Profile

En la pantalla de edición de perfil se nos muestra la información de nuestro usuario en campos editables para poder sobrescribir la información.

Carta Usuario



Al darle al botón de “Show profile ” o “Muestra perfil” se te desplegara una pantalla mostrándote tanto tu edad como tu nombre de usuario y tanto los lenguajes que conozcan, se te mostrara como te ven las demás personas a ti, así, si no te gusta algún punto de tu presentación a los usuarios, lo podrás cambiar detenidamente



Futuras Mejoras

- Envío de Audios e Imágenes a través de los chats.
- Restablecimiento de contraseña a través de un código de seguridad a través de correo del usuario.
- Mejora de la gestión de los profesores
- Funcionalidad Premium mejorarla
- Búsqueda de Chats y Profesores
- Verificación de identidad de los Profesores
- Funcionalidades Extra a los Profesores

Conclusión

Love 4 Code es un proyecto viable y prometedor que ha demostrado su potencial para conectar a programadores y profesionales del sector tecnológico. Aunque el desarrollo inicial ha sido desafiante, el equipo ha logrado superar los obstáculos y entregar una aplicación funcional con un enfoque claro en la experiencia del usuario y la escalabilidad. Con una planificación adecuada y una ejecución eficiente, este proyecto tiene el potencial de convertirse en una herramienta valiosa para la comunidad de programadores, fomentando tanto conexiones profesionales como personales.