Proyecto de Taller de Joyería – Avance del Proyecto Segundo Parcial

Modelos y Documentación del Software

Integrantes:

• Laura Valentina rozzo espinel

• Yudith Daniela Triana Vergel

• Andres Felipe Gutierrez Reyes

Docente:

Ing. Hely Suarez Marin

Fecha:

Octubre / 7 / 2025

FESC

## Problema

En la actualidad, la búsqueda de relaciones personales significativas se ve dificultada por la alta cantidad de interacciones superficiales en línea, la falta de herramientas fiables para verificar identidades y la exposición a fraudes o perfiles falsos. Muchas aplicaciones priorizan la cantidad sobre la calidad, lo que genera experiencias frustrantes, sensación de inseguridad y pérdida de confianza entre quienes buscan parejas o amistades auténticas. Se requiere una plataforma que facilite encuentros genuinos, proteja a sus usuarios y ofrezca resultados relevantes y verificables.

## Solución

Desarrollar una aplicación móvil de citas centrada en seguridad, veracidad y compatibilidad. La plataforma combinará verificación obligatoria de identidad (documentos + reconocimiento facial con prueba de vida), un algoritmo de compatibilidad que valore personalidad e intereses, y mecanismos de privacidad y reporte robustos.

## Actores

 **Usuario** (persona que busca parejas): crea perfil, busca, da like, chatea, reporta, bloquea.

 **Administrador**: gestión de usuarios, revisión de reports, control del sistema.

 **Moderador**: revisa reports y mensajes marcados (si aplica).

 **Sistema de Verificación**: actor técnico (servicio) que valida identidad (correo/teléfono/ID opcional).

 **Algoritmo de Emparejamiento**: servicio que calcula coincidencias.

 **Servicio de Mensajería / Chat**: actor técnico que maneja mensajes cifrados y notificaciones.

 **Pasarela de Pago** (opcional): para suscripciones/pagos.

 **Sistema de Notificaciones**: push/email.

## Reglas clave del sistema

* Registro requiere verificación (correo o SMS); sin verificación no se puede usar chat ni dar likes.
* Emparejamiento sólo muestra perfiles que acepten ser visibles y que cumplan filtros (edad/ubicación).
* Mensajería va cifrada (end-to-end si es posible) y los mensajes con denuncias quedan retenidos para moderación.
* Un usuario puede bloquear o reportar; el sistema registra acciones con fecha/hora/actor.
* Límites anti-spam: número de mensajes diarios a no-conexiones, límites para likes/swipes.
* Reglas de privacidad: no se exponen datos sensibles (teléfonos) salvo permiso explícito.
* Si verificación falla repetidamente, se suspende la cuenta hasta revisión manual.

## Funciones principales (lista estándar para app de citas)

* Registro / Login / Verificación (correo/SMS / opcional: verificación por documento)
* Crear/Editar perfil (foto, bio, intereses, preferencias)
* Búsqueda/Explorar (filtros por ubicación, edad, intereses)
* Emparejamiento automático (algoritmo por intereses/afinidad)
* Like / Super-like / Swipe (acción de interés)
* Match (cuando ambos se gustan)
* Chat seguro (mensajes, enviar fotos, eliminar conversación)
* Reportar / Bloquear usuario
* Notificaciones push / email
* Administración (gestion usuarios, revisar reports)
* Estadísticas básicas (matches, mensajes, actividad)
* (Opcional) Suscripción premium, boost de perfil

## Alcance (MVP)

El MVP concentrará las funciones esenciales para demostrar valor y seguridad:

\* Registro de usuario con datos básicos.

\* Verificación de identidad (documento + selfie con prueba de vida).

\* Creación y edición de perfil (bio, intereses, fotos).

\* Algoritmo básico de emparejamiento (filtros: edad, distancia, intereses).

\* Sistema de match (mutua aceptación) y chat cifrado entre matches.

\* Funcionalidad para reportar y bloquear usuarios.

\* Panel administrativo para revisar reportes y gestionar verificaciones.

## Diagrmas UML

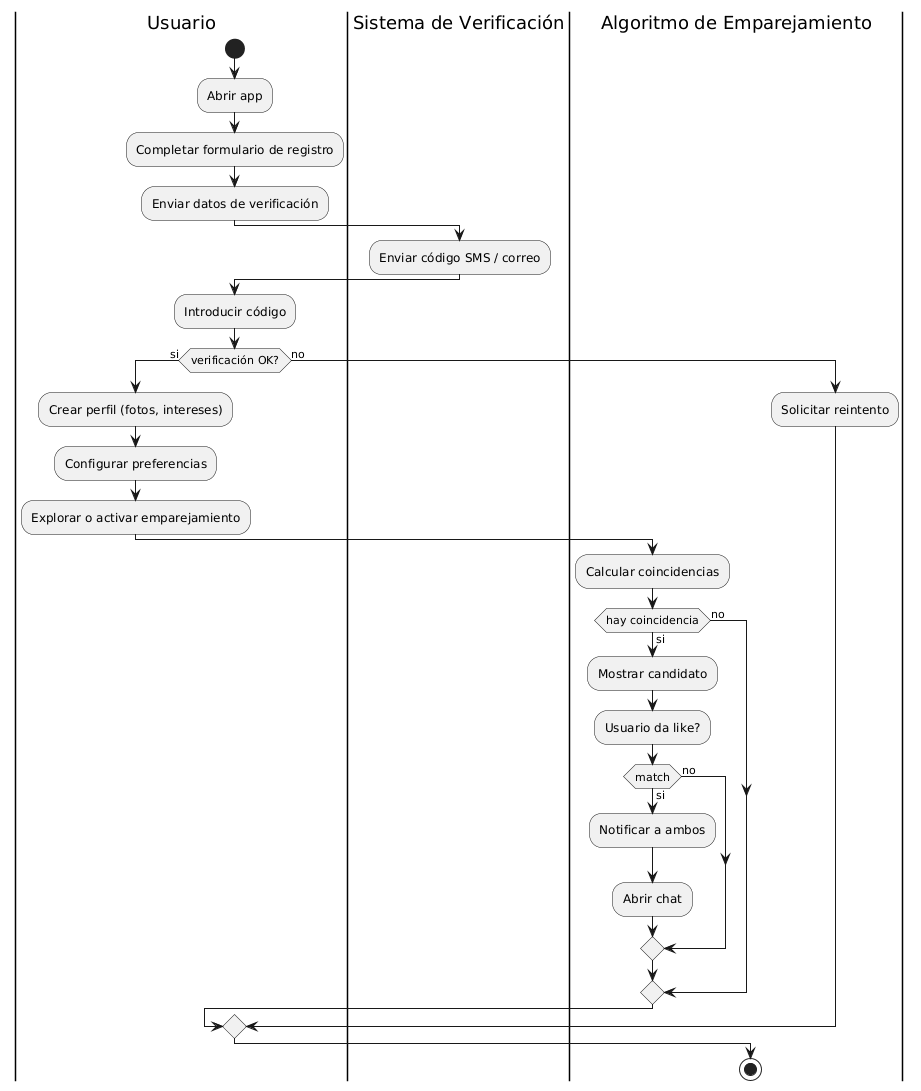
* Casos de uso:

Diagrama

El contenido generado por IA puede ser incorrecto.

El diagrama de casos de uso muestra las interacciones entre los actores externos (usuarios, administrador, moderador, servicios externos) y las funciones principales del sistema.  
Representa qué hace la app, no cómo lo hace.

* Diagrama de Actividades:



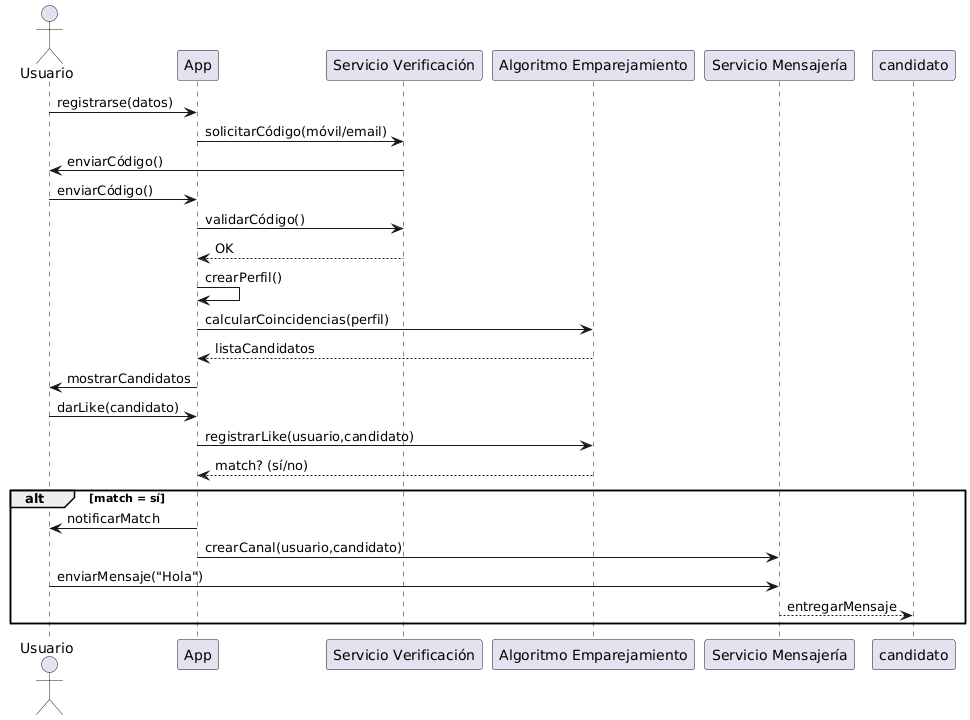
Representa el **flujo de trabajo o proceso** dentro de la aplicación, mostrando decisiones, bifurcaciones y condiciones.

En este caso, el diagrama de actividades describe **el proceso desde que un usuario se registra hasta que consigue un match**:

1. Registro y verificación.
2. Creación del perfil e intereses.
3. Búsqueda o emparejamiento automático.
4. Envío de “like”.
5. Si hay coincidencia, se abre el chat.

Este diagrama evidencia el flujo lógico del sistema, mostrando decisiones como “¿verificación exitosa?” o “¿hay match?”.

* Diagrama de Secuencia:

Muestra cómo los objetos o actores se comunican entre sí en el tiempo mediante mensajes.

El diagrama de secuencia de la App de Citas representa la interacción entre Usuario, Servidor y Algoritmo de Emparejamiento.  
Ejemplo:

1. El usuario se registra.
2. El sistema envía un código de verificación.
3. El usuario lo valida.
4. El sistema calcula coincidencias.
5. Si hay un match, ambos usuarios son notificados.

Permite visualizar el orden temporal de las operaciones y cómo se pasa la información entre los componentes.

* Diagrama de Comunicación: Tabla

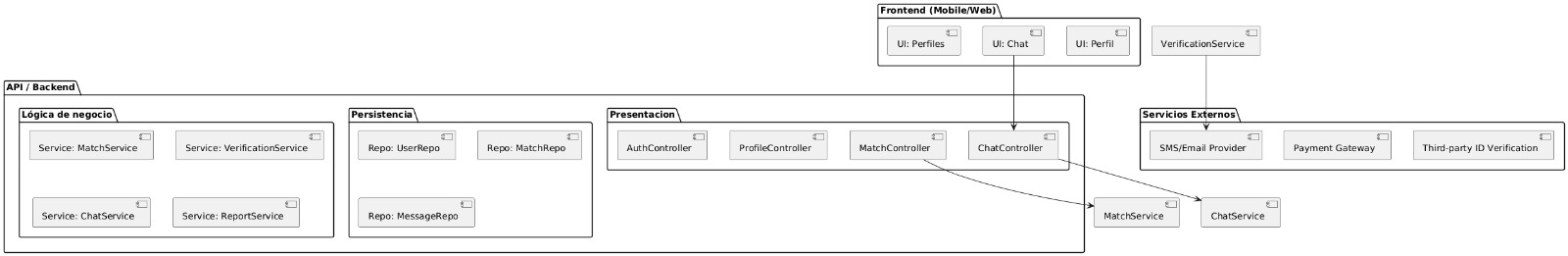
  El contenido generado por IA puede ser incorrecto.

Muestra cómo los objetos o actores se comunican entre sí en el tiempo mediante mensajes.

El diagrama de secuencia de la App de Citas representa la interacción entre Usuario, Servidor y Algoritmo de Emparejamiento.  
Ejemplo:

1. El usuario se registra.
2. El sistema envía un código de verificación.
3. El usuario lo valida.
4. El sistema calcula coincidencias.
5. Si hay un match, ambos usuarios son notificados.

Permite visualizar el orden temporal de las operaciones y cómo se pasa la información entre los componente

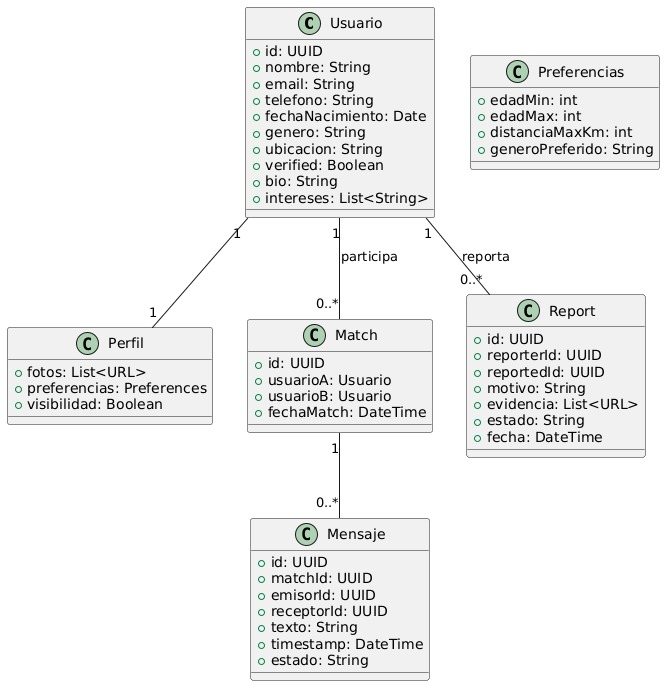
* Diagrama de paquetes:

Organiza el sistema en módulos o capas lógicas para mostrar cómo se agrupan las clases y componentes.

💬 Explicación

En la App de Citas se divide en:

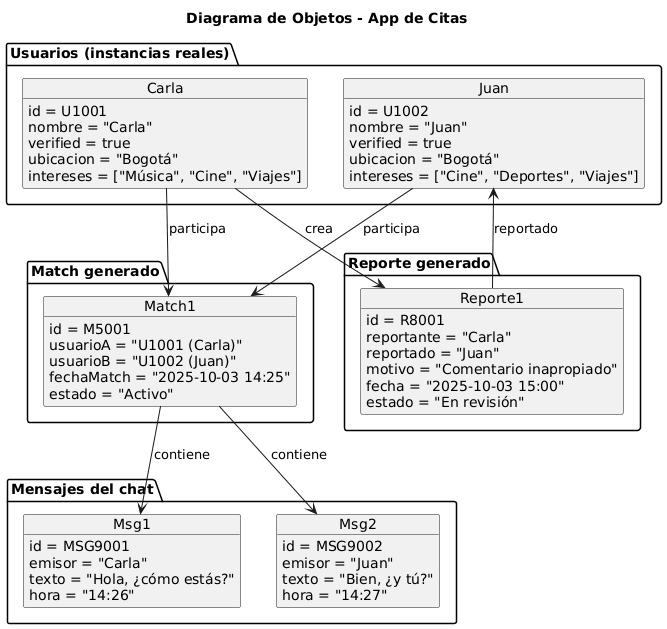
* Capa de presentación: interfaz de usuario móvil/web.
* Capa de lógica de negocio: algoritmos de emparejamiento, verificación y chat.
* Capa de datos: base de datos de usuarios, mensajes, matches y reportes.
* Servicios externos: API de SMS, correo o pasarela de pagos
* Diagrama de clases:



Define la estructura estática del sistema: clases, atributos, métodos y relaciones (asociación, herencia, composición).

En la App de Citas se representan clases como:

* Usuario, Perfil, Match, Mensaje, Reporte.  
  Cada una con sus atributos principales (nombre, edad, intereses, estado, fecha).  
  El diagrama muestra cómo los objetos se relacionan:
* Un Usuario puede tener varios Matches.
* Cada Match contiene varios Mensajes.
* Un Usuario puede generar Reportes.
* Sirve como modelo conceptual y base para el diseño de la base de datos.
* Diagrama de objetos:



* Diagrama de estados: Diagrama

  El contenido generado por IA puede ser incorrecto.

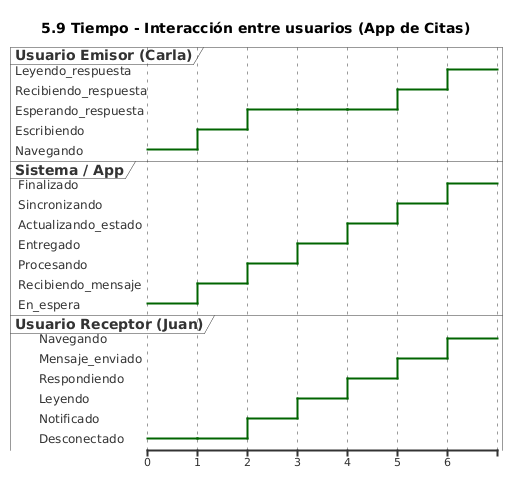
Describe el **ciclo de vida de un objeto**, mostrando los posibles **estados** por los que pasa y los **eventos** que causan los cambios.

Para la App de Citas, el ejemplo más claro es el ciclo de un **Match**:

1. No existe →
2. Usuario A da like →
3. Usuario B da like →
4. Match activo →
5. Chat iniciado →
6. Reportado o finalizado.

Permite comprender cómo se controlan los **flujos de cambio de estado** dentro del sistema.

* Diagrama de tiempo:



Muestra cómo cambian los estados de varios objetos o actores a lo largo del tiempo.

En la App de Citas, representa los estados de Carla, el Sistema y Juan desde el envío de un mensaje hasta su lectura:

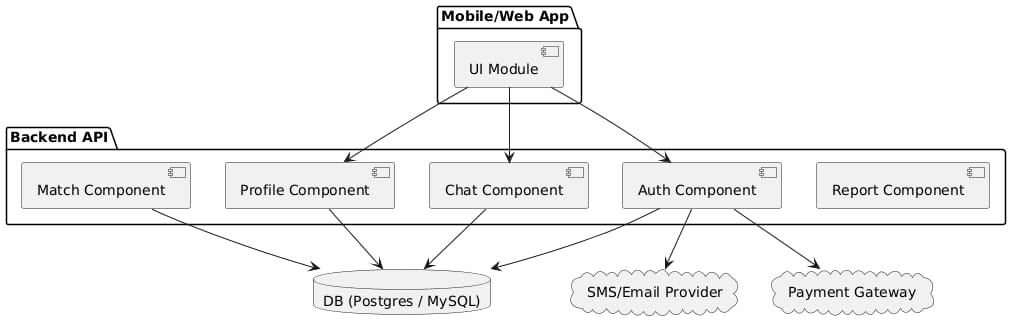
Carla: Escribiendo → Esperando → Leyendo respuesta.

Sistema: Recibiendo → Procesando → Entregado → Finalizado.

Juan: Desconectado → Notificado → Leyendo → Respondiendo.

Sirve para ilustrar la sincronización temporal y los momentos clave del intercambio.

* Diagrama de componentes:



Modela los elementos físicos o de software del sistema (módulos, APIs, bases de datos, servicios).

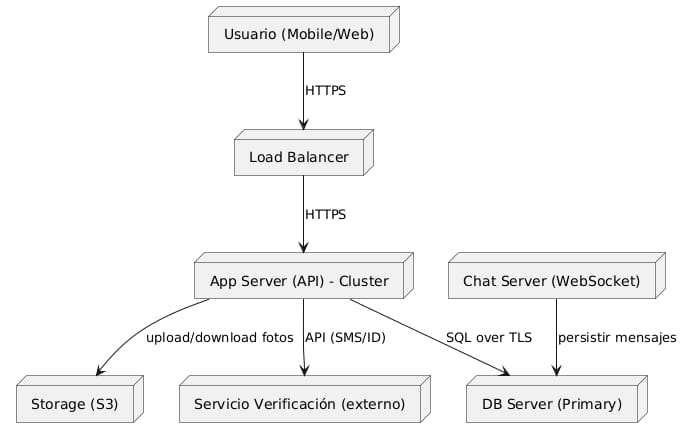
💬 Explicación

En la App de Citas:

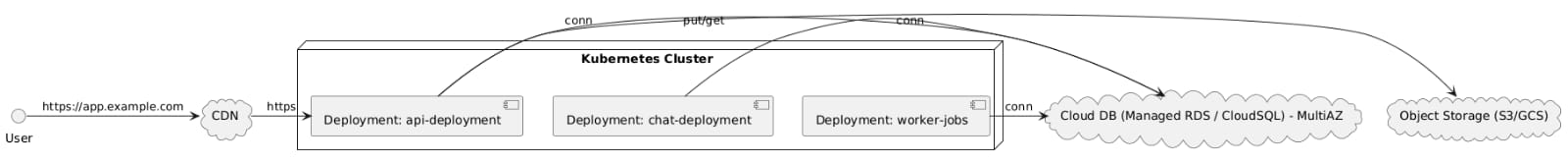
* Frontend móvil/web (interfaz).
* API principal (módulos de usuarios, matches, chat, reportes).
* Base de datos (MySQL o PostgreSQL).
* Servicios externos (verificación SMS, correo, almacenamiento de imágenes).

Este diagrama muestra cómo se conectan los componentes de software para que el sistema funcione integralmente.

* Diagrama de instalación:



* Diagrama de despliegue:



Muestra cómo se implementan y ejecutan los componentes del software en el entorno de producción (nodos, contenedores, redes).

En la App de Citas:

* Nodo Frontend (Web/Mobile): interfaz que consume la API.
* Nodo Backend (Servidor API): maneja la lógica de negocio.
* Nodo Base de datos (RDS o MySQL): almacena toda la información.
* Nodo Servicios externos (verificación, notificaciones, almacenamiento).

Representa la topología de ejecución del sistema, mostrando los nodos físicos y las conexiones entre ellos.