

ANTONIO DAVID MUÑOZ LEAL

ANALISIS DE PROYECTO



INDICE

INTRODUCCIÓN

1.1 Propósito	pág.3
1.2 Objetivo	
1.3 Viabilidad técnica y económica del proyecto	
1.4 Temporalización	pág.4

ANÁLISIS

2.1 Documentación relevante	pág.6
2.2 Definiciones	
2.3 Requisitos funcionales y no funcionales	pág.7
2.4 Casos de USO. Diagramas UML	pág.10
2.5 MODELO E/R. Diagramas E/R	pág.11

DISEÑO

3.1 Arquitectura hardware y software de la solución	
3.2 Modelado funcional. Diagrama de clase	pág.12.
3.3 Modelado de datos. Modelo relacional.	pág.13

IMPLEMENTACIÓN

4.1 Requisitos de instalación	pág.17
4.2 Implementación funcional. Clases	pág.18
4.3 Implementación de modelo de datos. Tablas	pág.21

PRUEBAS

5.1 Pruebas de módulos	pág.21
5.2 Pruebas de integración	pág.25
5.3 Pruebas del sistema	
5.4 Pruebas de instalación	pág.26

CONCLUSIONES

6.1 Grado de consecución	
6.2 Dificultades encontradas	pág.27
6.3 Propuestas de mejoras	
7 Bibliografía y recursos	pág.28

1.1 Propósito

Es un software que permite a usuarios comprar y vender productos entre sí, el usuario podrá subir a la aplicación cualquiera cosa que tenga en casa que ya no utiliza para venderla, éste subirá la foto de su producto y precio, lo anunciará en la propia aplicación, por lo cual los demás usuarios que tengan la necesidad de buscar un producto en específico, podrán darle una segunda vida a un producto olvidado.

El interesado por el producto podrá hacerle una oferta al dueño de este producto, y éste aceptará o no esta oferta. Luego como demás aplicaciones de compra/venta tanto comprador como vendedor, tendrán la posibilidad de compartir mensajes por un chat que existe entre los interesados y el anuncio al respecto.

La necesidad de crear este proyecto recae en la escasa comunicación en aplicaciones de este tipo, buscas un producto, ves su precio quizás no te interese y “te andas por las ramas” con el vendedor para intentar bajar el precio y comprarlo... En esta aplicación, tendrás la posibilidad de poder comprar el producto directamente, o lanzar una oferta hacia ese producto, una vez hecho esto, el propietario sin necesidad de mantener mensajes contigo podrá aceptar esta oferta y se realizaría el pago correspondiente por parte del interesado. Con esto nos hemos ahorrado el paso de tener un previo contacto con el vendedor, vas directamente a lo que te interesa, hacer una oferta y llevarte ese producto por el mejor precio posible.

En la vida cotidiana podríamos decir que esta aplicación es una mezcla entre una famosa aplicación de venta de segunda mano como por ejemplo Wallapop, que ves anuncios y contactas con el propietario para interesarte, y a su vez por otro lado como la aplicación de venta de ropa de segunda mano como Vinted, en la que un gran porcentaje de las ventas, se realizan de ofertas que mandan sus interesados, estos anuncios tienen un precio y tu como comprador sólo necesitas mandar una oferta al dueño.

1.2 Objetivos

El objetivo principal es quitar esa capa intermedia que existe entre la primera vez que ves un anuncio que te interesa y la compra de éste. Si ese anuncio te gusta, y crees que su precio es el correcto, puedes comprarlo directamente, pero si quieres pagar menos por él, lánzale una oferta al propietario, no necesitas hablar con él para nada. El propietario ya se encargará de aceptarla si le interesa.

Así se aligeran las compras que se realicen dentro de esta aplicación, que esto a su vez significa más movimiento de dinero y más ingresos para la empresa ficticia. Mientras que en una conversación normal entre interesados pueden pasar horas e incluso días, aquí sólo la espera de un click para aceptar la oferta.

1.3 Viabilidad técnica y económica

Continuando con el tema económico, esta aplicación sin contar con su precio de coste y puesta a producción. Tendríamos que contar con trabajadores que sean expertos en diferentes categorías que existan en nuestra aplicación, ya que necesariamente tendrán que moderar los anuncios que los clientes suben. Estos pueden o no cumplir las normas de la comunidad Selling 4 Us.

Y con los administradores que dan de alta a los usuarios moderadores y le asignan una categoría, o crear nuevos tipos de restricciones para futuras normas.

1.4 Temporalización

El problema principal de este proyecto es su extensión y complejidad en algunas partes de su utilidad. En la aplicación de escritorio para moderadores los anuncios que reciben estos moderadores, se actualizan y le llegan de manera automática al moderador, en un caso, existen 3 moderadores de la categoría Hogar en línea, y hay una alta cantidad de anuncios que revisar, estos irán trabajando e investigando si cumplen las normativas de la empresa, mientras que, en segundo plano, el servidor irá buscando si existen nuevos anuncios para que ellos puedan

moderar. Por lo tanto, si los moderadores no tienen anuncios tendrán que esperar a que el servidor, les mande nuevos, esto se hace de manera automática y se van repartiendo los anuncios entre moderadores, para que la carga de trabajo sea igual para todos.

Por otro lado, la aplicación Android que es la que utiliza el usuario Estándar, como puntos fuertes tiene la compra, oferta, chat y revisión de restricciones. Todas estas cuentan con avisos y notificaciones a tiempo real si son necesarias, para ello me ayudo de la base de datos ya que, si existen cambios o nuevas notificaciones que revisar, tendrá que ser guardada en memoria permanente, para la siguiente vez que se conecte el usuario.

Si el usuario está conectado y utilizando su aplicación le saldrán alertas y notificaciones en pantallas mostrando las nuevas interacciones a revisar.

Entonces teniendo en cuenta la lógica del servidor y sus moderadores, y la complejidad entre notificaciones y comunicaciones entre clientes móviles, han hecho retrasar la finalización de este proyecto, llegando a como, por ejemplo, requisitos como valoraciones de compradores, que salió como nuevo requisito en una revisión no se ha llegado a implementar al completo.

2. Análisis

2.1 Documentación relevante

Dentro de este apartado como clave a destacar, se utiliza en varias partes de la aplicación geolocalización de tanto los anuncios como el moderador. Por ejemplo, la aplicación móvil el cliente podrá filtrar por localidades cercanas anuncios que le interesen, Kayak en Cádiz y alrededores en un rango de precio entre 80€ y 190€. Para esto me ayudo con la geolocalización del usuario, para mostrar un mapa y marcar las ciudades cercanas y si éste desea seleccionar un rango para el filtrado. Google maps proporciona una API para ayudarte con la geolocalización y demás, tiene un método en este api que es searchbynear, lo que significa que te trae lugares cercanos a tu ubicación, pero con la desventaja que sólo trae 1 punto (El más cercano). Para solventar este problema me he

provisionado de un fichero csv con el nombre, latitud y longitud de todas las localidades de España. Este documento ha sido extraído de webs oficiales nacionales. Por lo que la información es verídica. Lo cual he tenido que crear mi propia API, dado un punto exacto, traerme 15 localidades cercanas (Ayudándome de una fórmula matemática con el radio de la tierra) a mi ubicación, y así poder mostrarlas en el mapa para seleccionar y filtrar.

Además de consultas técnicas en páginas como StackOverFlow, Oracle Java, developer Android etc.

2.2 Definiciones

Administrador: Entidad que como su nombre lo refiere, administra el sistema, añadiendo o eliminando moderadores y restricciones.

Moderador: Entidad que depende de una categoría de anuncios, que es experto en ella, y controla que se cumplen las normativas de la aplicación y da validez o no a los anuncios subidos por usuarios Estándares.

Usuario Estándar: Son los que le dan la razón de ser a esta aplicación, suben anuncios, compran, venden, lanzas ofertas, las reciben aceptan o deniegan, se comunican entre ellos a tiempo real con el chat, buscan nuevos anuncios filtran por localización, categoría rango de precios, revisan sus anuncios bloqueados etc.

2.3 Definiciones

Funcionalidades principales

Crear, modificar y gestionar tus propios anuncios, buscar y listarlos tanto por ubicación o por palabras claves, crear ofertas a diferentes al igual que comprar y comunicación entre comprador y vendedor mediante chat. Además de una implementación de gestión de pagos propia (PayPal).

Roles

Cada usuario tendrá su rol estándar (Comprador/Vendedor) ya que el mismo usuario podrá comprar y vender. Un usuario tendrá una valoración para ayudar a los compradores a orientarse. Con este rol le permitirá poder subir anuncios, buscar, poder chatear, etc....

Se implementa un rol moderador en el que, desde la aplicación de escritorio, tendrá una lista de anuncios que irán apareciendo de forma automática según la categoría de este moderador, este moderador podrá bloquear un anuncio NO válido, ejemplo como no subir una imagen de contenido explícito, imagen ofensiva, palabras mal sonantes o soeces dentro del título o descripción del artículo. Por lo tanto, al momento que un moderador bloquee un anuncio, el dueño de este anuncio recibirá un mensaje en su aplicación móvil Este moderador será especialista en un tipo de categoría.

Además de un rol administrador que podrá crear usuarios moderadores y tipos de restricciones para estos

Tareas

La aplicación cliente de escritorio estará diseñada para que el usuario moderador pueda hacer log, y por ejemplo cada moderador tendrá asignado una localidad, este podrá desde la aplicación de escritorio ver una lista de todos los anuncios que existen actualmente en esa localidad, este usuario al ser encargado de confirmar que el producto es válido podrá rechazarlo enviándole un comentario al dueño, estos comentarios serán siempre los mismos, imagen, palabras soeces, etc.

La aplicación móvil será utilizada por los usuarios con rol estándar, comprador/vendedor. Estos, al momento de querer comprar un producto, tendrán una lista de anuncios filtrados por una localización, además de poder buscarlo por palabras claves y/o categorías o cerca de su ubicación.

Si al usuario le interese este producto, podrá chatear con el vendedor, enviarle una oferta, o directamente comprarle el producto. Estos se comunicarán por un chat para el envío. Además de que el comprador podrá valorar al vendedor

Si el usuario quiere vender un producto, lo subirá con imagen, título, descripción y localización. Este tendrá un apartado donde tendrá una lista de sus productos subidos.

Ambos usuarios tendrán una lista de chats con otros compradores/vendedores sobre un producto.

Servidor

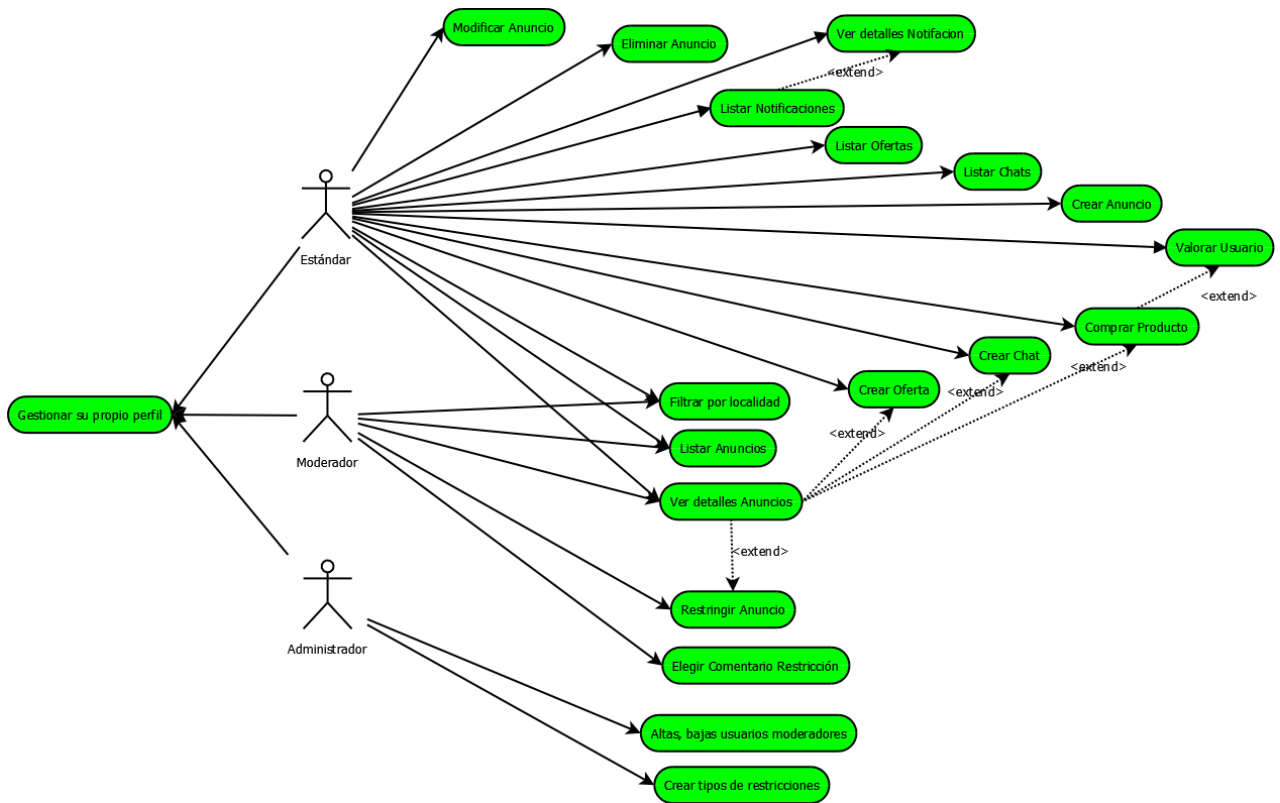
Gestionará la comunicación con los clientes, además de gestionar las llamadas a la base de datos (Hibernate).

Requisitos No funcionales

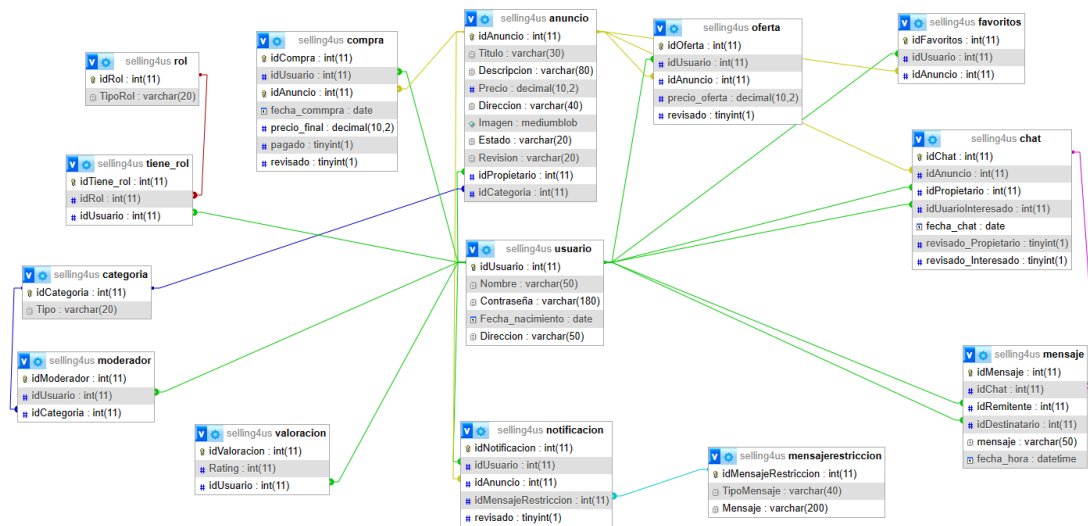
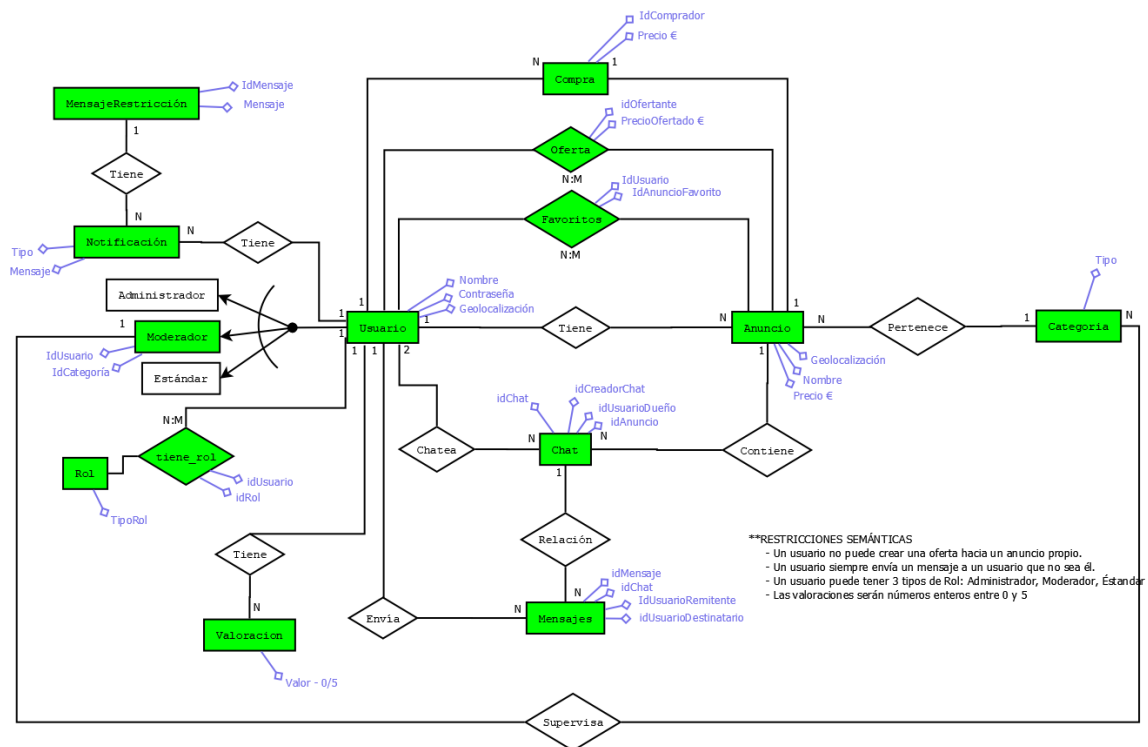
Escritorio: JDK 17, Windows 10, base de datos relaciones, internet

Móvil: Versión SDK 24, geolocalización y conexión a internet

2.4 Casos de uso



2.5 Modelado E/R. Diagramas E/R



3. Diseño

3.1 Arquitectura hardware y software de solución

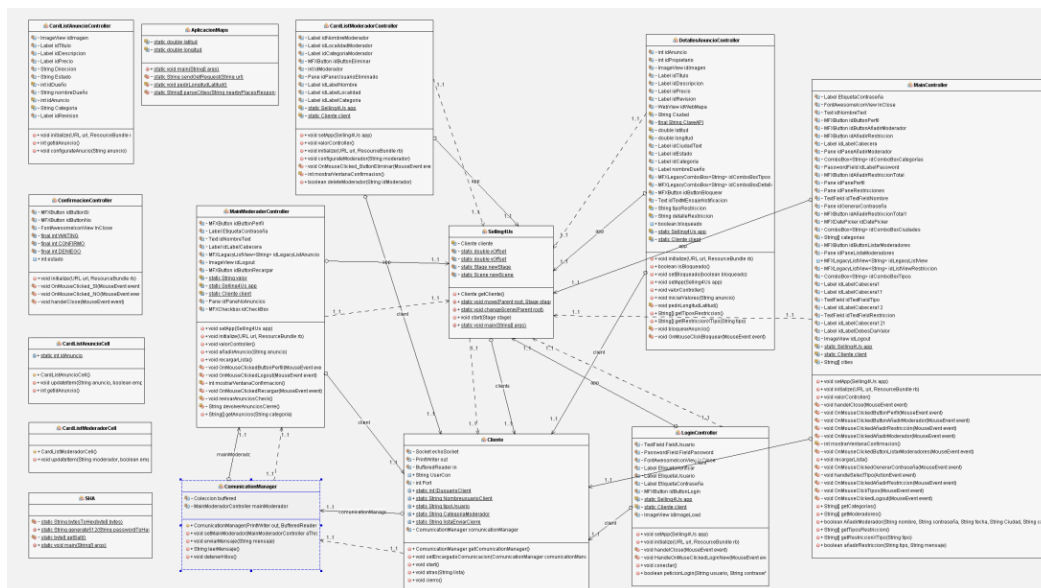
Servidor: Windows 7, Java17, Base de datos MYSQL, 2GB RAM, CPU Intel

Escritorio: Windows 7, Java17, 2GB RAM, CPU Intel

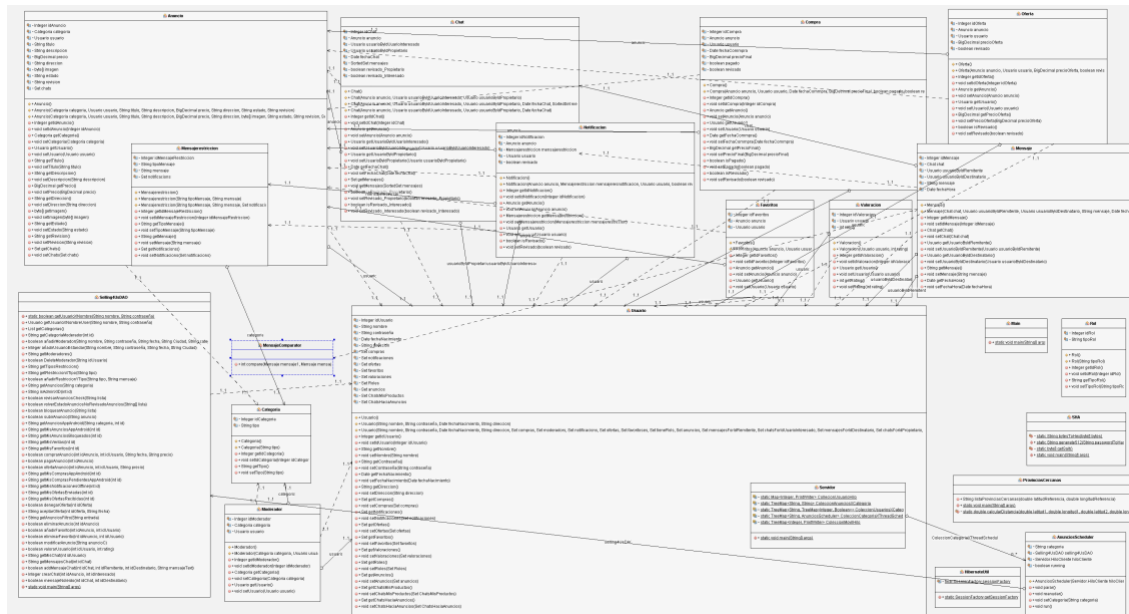
Android: SDK 24, permisos de conexión a internet, cámara y geolocalización

3.2 Modelado funcional de la solución. Diagrama de Clases

Servidor



Cliente



3.3 Modelado de Datos. Diccionario de datos

```
CREATE TABLE ANUNCIO(
idAnuncio int AUTO_INCREMENT,
Titulo varchar(30) NOT NULL,
Descripcion varchar(80) NOT NULL,
Precio DECIMAL(10,2) NOT NULL,
Direccion varchar(40) NOT NULL,
Imagen BLOB,
Estado varchar(20) NOT NULL,
Revision varchar(20) NOT NULL,
idPropietario int NOT NULL,
idCategoria int NOT NULL,
PRIMARY KEY (idAnuncio),
FOREIGN KEY (idPropietario) REFERENCES USUARIO (idUserario) ON DELETE CASCADE,
FOREIGN KEY (idCategoria) REFERENCES CATEGORIA (idCategoria)
);
```

```
CREATE TABLE USUARIO(  
idUsuario int AUTO_INCREMENT,  
Nombre varchar(50) NOT NULL,  
Contraseña varchar(180) NOT NULL,  
Fecha_nacimiento date NOT NULL,  
Direccion varchar(50) NOT NULL,  
PRIMARY KEY(idUsuario)  
);
```

```
CREATE TABLE ROL(  
idRol int AUTO_INCREMENT,  
TipoRol varchar(20) NOT NULL,  
PRIMARY KEY(idRol)  
);  
  
CREATE TABLE TIENE_ROL(  
idTiene_rol int AUTO_INCREMENT,  
idRol int NOT NULL,  
idUsuario int NOT NULL,  
PRIMARY KEY (idTiene_rol),  
FOREIGN KEY (idRol) REFERENCES ROL (idRol),  
FOREIGN KEY (idUsuario) REFERENCES USUARIO (idUsuario)  
);
```

```
CREATE TABLE MODERADOR(  
idModerador int AUTO_INCREMENT,  
idUsuario int NOT NULL,  
idCategoria INT NOT NULL,  
PRIMARY KEY (idModerador),  
FOREIGN KEY (idCategoria) REFERENCES CATEGORIA (idCategoria),  
FOREIGN KEY (idUsuario) REFERENCES USUARIO (idUsuario)  
);
```

```
CREATE TABLE CATEGORIA(  
idCategoria int AUTO_INCREMENT,  
Tipo varchar(20) NOT NULL,  
PRIMARY KEY (idCategoria)  
);
```

```
CREATE TABLE FAVORITOS(  
idFavoritos int AUTO_INCREMENT,  
idUserario int NOT NULL,  
idAnuncio int NOT NULL,  
FOREIGN KEY (idUserario) REFERENCES USUARIO (idUserario) ON DELETE CASCADE,  
FOREIGN KEY (idAnuncio) REFERENCES ANUNCIO (idAnuncio) ON DELETE CASCADE,  
PRIMARY KEY(idFavoritos)  
);
```

```
CREATE TABLE OFERTA(  
idOferta int AUTO_INCREMENT,  
idUserario int NOT NULL,  
idAnuncio int NOT NULL,  
precio_oferta DECIMAL(10,2) NOT NULL,  
revisado BOOLEAN NOT NULL,  
FOREIGN KEY (idUserario) REFERENCES USUARIO (idUserario) ON DELETE CASCADE,  
FOREIGN KEY (idAnuncio) REFERENCES ANUNCIO (idAnuncio) ON DELETE CASCADE,  
PRIMARY KEY(idOferta)  
);
```

```
CREATE TABLE COMPRA(  
idCompra int AUTO_INCREMENT,  
idUserario int NOT NULL,  
idAnuncio int NOT NULL UNIQUE,  
fecha_compra date NOT NULL,  
precio_final DECIMAL(10,2) NOT NULL,  
pagado BOOLEAN NOT NULL,  
revisado BOOLEAN NOT NULL,  
FOREIGN KEY (idUserario) REFERENCES USUARIO (idUserario) ON DELETE CASCADE,  
FOREIGN KEY (idAnuncio) REFERENCES ANUNCIO (idAnuncio) ON DELETE CASCADE,  
PRIMARY KEY(idCompra)  
);
```

```

CREATE TABLE CHAT(
idChat int AUTO_INCREMENT,
idAnuncio int NOT NULL,
idPropietario int NOT NULL,
idUserioInteresado int NOT NULL,
fecha_chat date,
revisado_Propietario BOOLEAN NOT NULL,
revisado_Interesado BOOLEAN NOT NULL,
PRIMARY KEY(idChat),
FOREIGN KEY (idAnuncio) REFERENCES ANUNCIO (idAnuncio),
FOREIGN KEY (idPropietario) REFERENCES USUARIO (idUserio) ON DELETE CASCADE,
FOREIGN KEY (idUserioInteresado) REFERENCES USUARIO (idUserio) ON DELETE CASCADE
);

CREATE TABLE MENSAJE(
idMensaje int AUTO_INCREMENT,
idChat int NOT NULL,
idRemitente int NOT NULL,
idDestinatario int NOT NULL,
mensaje varchar(50) NOT NULL,
fecha_hora DATETIME NOT NULL,
PRIMARY KEY (idMensaje),
FOREIGN KEY (idChat) REFERENCES CHAT(idChat),
FOREIGN KEY (idRemitente) REFERENCES USUARIO(idUsuario) ON DELETE CASCADE,
FOREIGN KEY (idDestinatario) REFERENCES USUARIO(idUsuario) ON DELETE CASCADE
);

```

```

CREATE TABLE MENSAJERESTRICCION(
idMensajeRestriccion int AUTO_INCREMENT,
TipoMensaje varchar(40) NOT NULL,
Mensaje varchar(200) NOT NULL,
PRIMARY KEY(idMensajeRestriccion)
);

CREATE TABLE NOTIFICACION(
idNotificacion int AUTO_INCREMENT,
idUserio int NOT NULL,
idAnuncio int NOT NULL,
idMensajeRestriccion int NOT NULL,
revisado BOOLEAN NOT NULL,
PRIMARY KEY(idNotificacion),
FOREIGN KEY (idUserio) REFERENCES USUARIO(idUsuario) ON DELETE CASCADE,
FOREIGN KEY (idAnuncio) REFERENCES ANUNCIO(idAnuncio) ON DELETE CASCADE,
FOREIGN KEY (idMensajeRestriccion) REFERENCES MENSAJERESTRICCION(idMensajeRestriccion)
);

```

```

CREATE TABLE VALORACION(
idValoracion int AUTO_INCREMENT,
Rating int NOT NULL,
idUserio int NOT NULL,
PRIMARY KEY(idValoracion),
FOREIGN KEY (idUserio) REFERENCES USUARIO (idUserio) ON DELETE CASCADE
);

```

4. Implementación

4.1 Requisitos de instalación y ejecución

Servidor

- Sistema Operativo Windows 7
- JDK 1.8
- Gestor de base de datos MYSQL
- El ejecutable .exe deberá contener el script SQL de creación de tablas con mínimo un usuario Administrador
- Fichero Config.init con la configuración de red
- Conexión a la red

Cliente

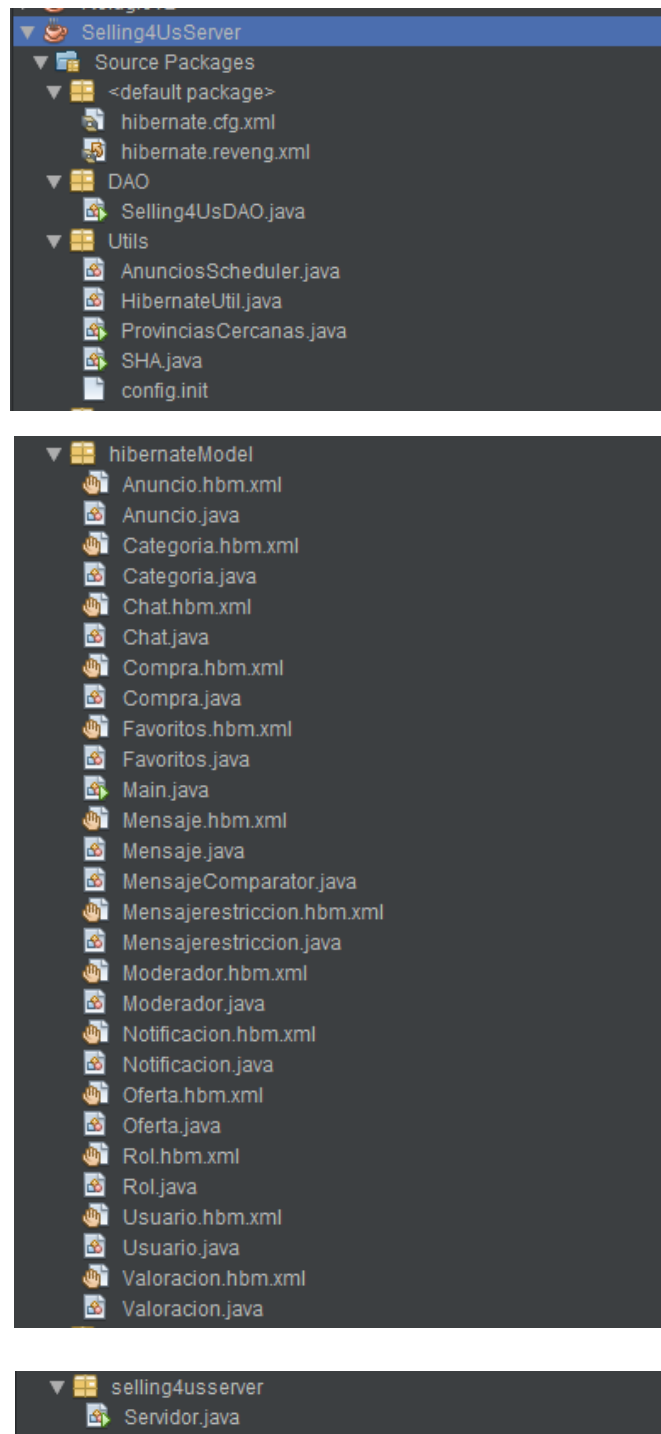
- Sistema Operativo Windows 10
- JDK 17
- Conexión a la red
- Librerías necesarias cargadas por ejemplo JavaFX
- Clave key válida para servicios de Google MAPS

Android

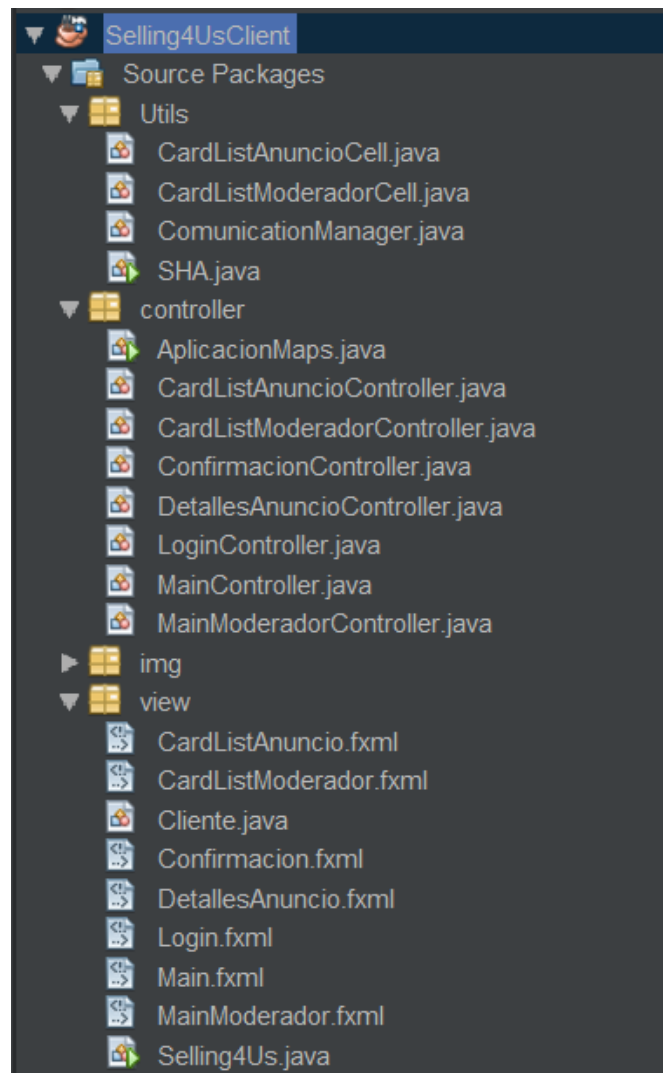
- SDK 24
- Mínimo necesario conexión a la red, y a la ubicación actual
- Clave key válida para servicios de Google MAPS

4.2 Implementación Funcional. Clases

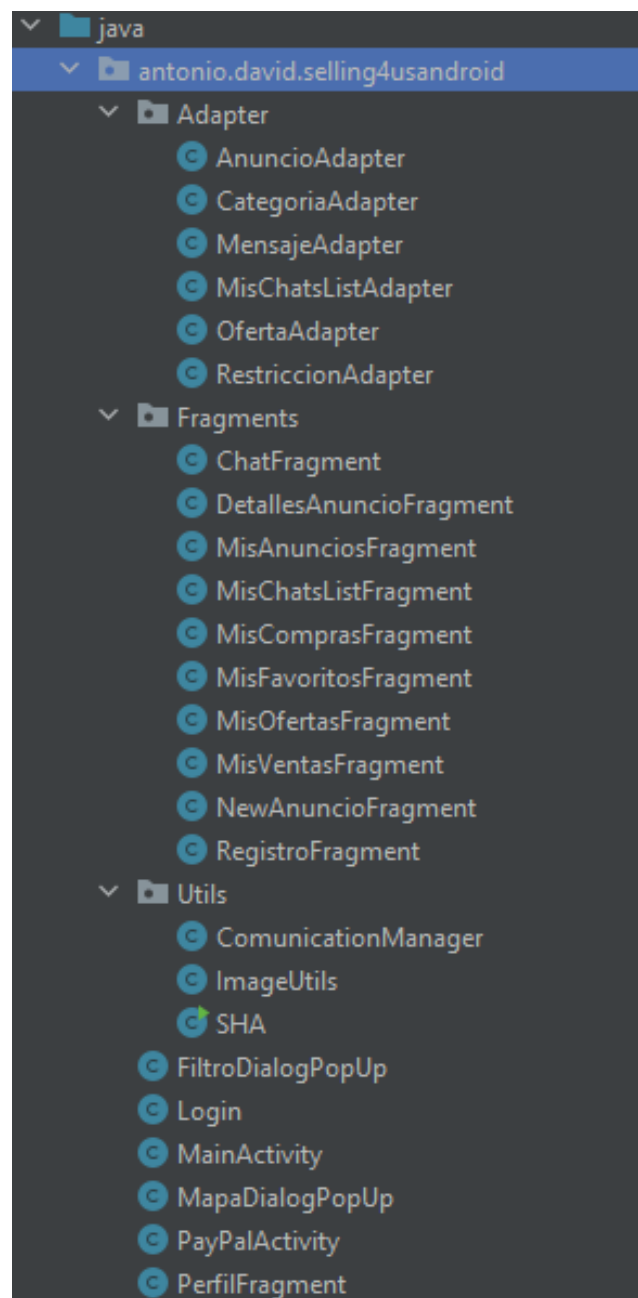
Servidor



Cliente Escritorio



Movil



4.3 Implementación del modelo de datos. Tablas

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> anuncio	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	18	InnoDB	utf8mb4_general_ci	7.3 MB
<input type="checkbox"/> categoria	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> chat	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	11	InnoDB	utf8mb4_general_ci	64.0 KB
<input type="checkbox"/> compra	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> favoritos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> mensaje	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	84	InnoDB	utf8mb4_general_ci	64.0 KB
<input type="checkbox"/> mensajerestriccion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> moderador	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> notificacion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	64.0 KB
<input type="checkbox"/> oferta	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> rol	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> tiene_rol	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8mb4_general_ci	48.0 KB
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> valoracion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_general_ci	48.0 KB

5. Pruebas

5.1 Pruebas de módulos

Servidor

Cliente Escritorio (Administrador):

- Login
- Añadir moderador: inserción de datos para añadir un nuevo moderador, dando su nombre, contraseña que puede generarse automáticamente, fecha de nacimiento, Dirección de residencia y categoría especializada.
- Eliminar moderador: Lista de todos los moderadores existentes, y la posibilidad de eliminarlo de la lista y a su vez, de la base de datos.
- Añadir Restricciones: Listado e inserción de nuevas normativas y restricciones para bloqueo de anuncios de usuarios Estándar

Cliente Escritorio (Moderador):

- Login
- Listado de Anuncios: El moderador recibirá automáticamente nuevos anuncios cuando sea necesario. Este moderador cuando inicie sesión, si existen anuncios aparece una lista de ellos, cuando revise todos, hará click en el botón revisar y si existen recibirá más anuncios, si no, permanecerá a la espera a que le lleguen nuevos anuncios automáticamente.
- Ver detalles Anuncio: Cada anuncio que aparece en el listado, contiene una información general de éste, el usuario moderador al hacer click en uno de ellos podrá ver en una ventana emergente con todos los detalles necesarios para comprobar el cumplimiento de las normativas de éste.
- Bloqueo de Anuncios: Una vez visto los detalles del anuncio, el usuario moderador, podrá desde el botón que existe dentro de detalles anuncio bloquear este anuncio, eligiendo previamente el tipo de restricción. Luego al volver a atrás a la lista de anuncios, el anuncio aparecerá deshabilitado ya que una vez bloqueado, el único usuario que puede desbloquearlo es el usuario Propietario cambiando los datos necesarios según la restricción.
- Revisar Anuncios: Una vez revisado todos los anuncios el moderador hará click en el botón revisar, y la lista de anuncios se actualizarán sus estados a Revisados o bloqueados.

Cliente Móvil (Usuario Estándar):

- Login: Introducir usuario y contraseña para iniciar sesión
- Registro: Introducir usuario, fecha de nacimiento, Dirección (Controlada por Geolocalización) y contraseña

- Listado de anuncios por categoría: El usuario en primera instancia podrá lista anuncios por su categoría.
- Filtrado de anuncios: La aplicación proporciona un panel de filtrado, en el que tienes que proporcionarle como mínimo un nombre de producto, por ejemplo, kayak. Si buscas sólo por nombre aparecerán todas las coincidencias que existan en la bbdd. Si quieres puedes filtrar por más campos como Ubicación, que el usuario abrirá el mapa pulsando el botón abrir mapa, y una vez dentro tendrá un círculo para poder seleccionar las localidades cercanas para filtrar.
- Ver detalles Anuncio: Una vez listados, el usuario podrá hacer click en alguno de ellos para poder ver más detalles de éste
- Añadir Anuncio a Favoritos: Una vez listados, el usuario podrá hacer una pulsación larga para añadir el anuncio a sus favoritos.
- Comprar Anuncio: Dentro de sus detalles aparecerá un botón de comprar anuncio que al pulsar en él le redirigirá a la pasarela de pago.
- Oferta Anuncio: Dentro de los detalles aparecerá un botón de oferta que mostrará un dialogo con el precio que quires ofertar el propietario.
- Abrir chat sobre un anuncio en específico: Dentro de los detalles aparecerá un botón de chat que abrirá un nuevo chat con su propietario

- Listado de favoritos: En el menú desplegable de la aplicación, pulsamos en el ítem Favoritos y nos listará nuestros anuncios favoritos.
- Eliminar de favoritos: una vez el anuncio sea favorito, podremos eliminarlo de favoritos, haciendo una pulsación larga y eliminamos de favoritos
- Listar mis Anuncios: En el menú desplegable de la aplicación, pulsamos en el ítem Mis anuncios donde aparecerán una lista de mis anuncios activos (No bloqueados o no vendidos) en la aplicación.
- Listar mis Anuncios Bloqueados: En el menú desplegable de la aplicación, pulsamos en el ítem Mis anuncios donde podremos dentro de esta pantalla pulsar en mis anuncios Bloqueados.
- Modificar Anuncio: Una vez listado los anuncios restringidos, veremos el tipo de restricción que tiene el anuncio. Al entrar en él, podremos modificarlo para que vuelva a ser revisado por el moderador.
- Listar Anuncios comprados: En el menú desplegable de la aplicación, pulsamos en el ítem Mis compras donde tendremos una lista de nuestros anuncios Comprados.
- Listar Ofertas recibidas: En el menú desplegable de la aplicación, pulsamos en el ítem Mis Ofertas.
- Aceptar o rechazar Oferta: En el listado de Ofertas, tenemos 2 botones, verde para aceptar y rojo para rechazar.
- Listar Chat usuario: En el menú desplegable de la aplicación, pulsamos en el ítem Mis Chats.

- Compartir mensajes chat: Dentro de un chat, pulsar botón enviar mensaje.

5.2 Pruebas de integración

Cliente escritorio:

- Login: Enviar Usuario y contraseña desde el cliente solo si los campos no están vacíos, y si el usuario es moderador, entrar en la parte de moderadores, y si es administrador hacia el administrador. Cambiar de estado en el protocolo del servidor desde LOGINREGISTRAR a RUNNIG.
- Administrador:
 - Enviar sólo al servidor, los valores de los campos del nuevo moderador, sólo cuando estos campos estén completos.
 - Enviar nueva restricción solo si se ha seleccionado un tipo de restricción concreto y el campo no está vacío.
- Moderador:
 - Recibir nuevos anuncios sólo si el usuario está a la espera de nuevos anuncios.
 - Pedir nuevos anuncios, sólo si los anuncios presentes están revisados.
 - Bloquear Anuncio sólo si se ha elegido una restricción específica.

Cliente Móvil:

- Login: Enviar Usuario y contraseña desde el cliente solo si los campos no están vacíos, Cambiar de estado en el protocolo del servidor desde LOGINREGISTRAR a RUNNIG.
- Registrar: Enviar campos del nuevo usuario sólo si los valores no están vacíos y las contraseñas coinciden.
- Eliminar Anuncio sólo si eres el propietario.
- Añadir a favoritos sólo si no eres propietario.
- Comprar Anuncio sólo si no eres propietario.
- Hacer Oferta Anuncio sólo si no eres propietario.
- Abrir chat a un Anuncio sólo si no eres propietario

- Filtrar por nombre y categoría sólo si los demás campos del panel de filtrado están vacíos.
- Filtrar por nombre, categoría y rango de precio sólo si existen los valores de precios no vacíos.
- Filtrar por todos los campos sólo si se han seleccionado localidades en el mapa.
- Mostrar el listado de anuncios sólo si existe alguno.

5.3 Pruebas de Sistema.

- Compatibilidad: Para que la aplicación sea compatible con todo tipo de dispositivos y lenguajes, los paquetes que se envían entre cliente y servidor, son cadenas tipo texto, separadas por caracteres específicos
- Rendimiento: El servidor lanza un hilo que atiende al cliente por cada conexión de nuevo cliente. Por lo tanto, su rendimiento y reescalabilidad son mayores.
- Seguridad: las contraseñas se encriptan antes de cada conexión.
- Interfaz de Usuario: Clara sencilla y sin estilos sobrepasados.

5.4 Pruebas de Instalación

- En el lado servidor, necesitamos que reciba bien la dirección IP y el puerto, la conexión a la base de datos, y que contenga las librerías para el mapeo Objeto-relacional con hibernate junto con la JDBC.
- En el lado cliente, necesitamos que contenga las librerías necesarias en la instalación y la dirección IP y puerto del servidor.
- En la aplicación Móvil, necesitamos que el dispositivo contenga la versión mínima de sistema operativo, acceso a la ubicación y a la cámara.

6. Conclusión

6.1 Grado de consecución de los objetivos inicialmente planteados

Se han trabajado en todos los requisitos funcionales y no funcionales planteados, atendiendo en gran parte del tiempo a nuevos problemas encontrados y sugeridos en revisiones periódicas.

6.2 Dificultades encontradas

Servidor

Nos encontramos principalmente problemas que pudimos resolver gratamente con la comunicación entre hilos para los anuncios de los moderadores. El servidor es la única entidad lógica que está en contacto con la base de datos, por lo cual es la responsable de otorgar datos a los clientes, en este caso clientes moderadores. En revisiones se pidió que los anuncios se mostraran y se repartieran entre moderadores automáticamente, esto quiere decir que, un moderador permanecerá a la espera de que le lleguen anuncios nuevos a revisar. Por lo tanto, en el servidor tengo varios mapas que me ayudan a saber que moderadores son los que están conectados, de que categoría específica y cuáles son los que están esperando un anuncio.

Para esto, creamos un hilo AnuncioScheduler que, si es necesario (Porque existen moderadores de una categoría específica a la espera de nuevos anuncios), hará peticiones cada 10 segundos a la bbdd en busca de nuevos anuncios no revisados de esta categoría.

Si encuentra nuevos anuncios, el hilo AnuncioScheduler añadirá la lista de estos anuncios a una colección compartida del Hilo del cliente, y notificará a este hilo para que sepa que es posible enviar nuevos anuncios al siguiente moderador, luego buscará el siguiente moderador posible en el mapa y le enviará el nuevo anuncio.

Cliente escritorio

En el cliente uno de los principales problemas, aunque no sea muy grave es la redimensión de la imagen del anuncio, con JavaFx no he conseguido que tenga un correcto encuadre en el marco necesario.

Cliente Móvil

Una de las dificultades que hemos conseguido salvar son las múltiples comprobaciones para las notificaciones en compras, oferta, bloqueo de anuncios y chats. Sobre todo, para chats ya que, tenemos en cuenta, si un usuario está online, si ese usuario estando online no está dentro de la conversación, también tenemos que tener en cuenta si ese usuario no está online, el chat pasara a un estado no revisado para que la próxima vez que se conecte le salga los nuevos cambios.

Otros

Tiempo, nuevas peticiones a implementar cada revisión, trabajar todos los fines de semana.

6.3 Propuesta de mejora y posibles ampliaciones

Mejora visual de colores y estilos en la interfaz de escritorio para administradores y moderadores. Pasarela de pago real con, y hostear el servidor para poder conectarme desde cualquier ubicación.

7. Bibliografía y recursos on-line

- Preguntas en foros
- Ayuda para ideas entre compañeros de clase
- Ayuda profesores
- Youtube
- Android for developers
- Inteligencia Artificial para creación de logo dudas en algunos conceptos para android