



Índice

1.Introducción.....	4
2.Plan de Empresa.....	5
2.1.Justificación.....	5
2.1.1.¿Por qué este negocio?.....	5
2.1.2.¿Existe Competencia?.....	5
2.1.3.¿Por qué motivos se venderá el producto?.....	5
2.1.4.¿Qué me diferencia de los demás?.....	6
2.1.5.¿Qué hay de novedoso en mi producto?.....	6
2.2.Nombre y Logo.....	7
2.3.Producto.....	8
2.3.1.¿Qué vas a ofrecer?.....	8
2.3.2.Finalidad.....	8
2.4.Modelo Negocio.....	9
2.5.Consumidores y Posibles Clientes.....	10
2.5.1.Tipos de Consumidores.....	10
2.5.2.Zona en la que viven.....	10
2.6.Competencia.....	10

2.7.DAFO.....	11
2.7.1.Debilidades.....	11
2.7.2.Amenazas.....	11
2.7.3.Fortalezas.....	12
2.7.4.Oportunidades.....	12
2.8.Publicidad y promoción.....	13
3.Descripción del sistema.....	14
3.1.Cliente.....	14
3.2.Administrador.....	15
3.3.Mejoras Futuras.....	15
4.Modelado.....	16
4.1.Sketching.....	16
4.2.Diagrama de Clases del modelo.....	17
5.Diseño.....	18
5.1.Diseño de la base de datos.....	20
6.Implementación.....	21

Introducción

El proyecto es una aplicación web centrada en dar información detallada sobre videojuegos y ofrecernos la posibilidad de saber si nuestro ordenador o portátil es adecuado para correr un juego deseado.

La aplicación solo tiene opciones de interacción para clientes pero no para administradores, esto se explicará más adelante.

El proyecto está desarrollado en lenguaje Kotlin y mostrado en Android.

2. Plan de Empresa

2.1. Justificación:

2.1.2. ¿Por qué este negocio?

Los temas de los videojuegos están abarcando cada vez más los medios de comunicación, tanto que mucha gente los usa como medio de ingreso o hasta de enseñanza en algunas escuelas, yo soy un gran consumidor de estos productos y siempre estoy al pendiente de cualquier novedad anunciada ya sea en las redes sociales o en televisión, por lo que se que una aplicación web relacionada con los videojuegos llamará la atención de muchos usuarios.

2.1.3. ¿Existe competencia?

Si existen aplicaciones que te informan sobre videojuegos, que dan información más amplia sobre juegos, los creadores y de sus empresas.

También hay otras que te informan sobre componentes para el pc o que brindan páginas recomendadas para comprar componentes en ofertas, etc.

2.1.4. ¿Por qué motivos creo que el producto se venderá?

Porque no hay aplicación que realice específicamente lo que hace mi aplicación y tampoco es que sea una aplicación muy complicada de usar, está pensado que sea muy intuitivo y de poco tiempo de carga para la comodidad del usuario.

2.1.5. ¿Qué me diferencia de los demás?

Mi aplicación además de informar que es lo que hace la mayoría de aplicaciones en la tienda, da más facilidades al usuario que normalmente da trabajo buscar por uno mismo y toma mucho tiempo.

2.1.6. ¿Qué hay de novedoso en mi producto?

Una persona interesada en los videojuegos al querer pillar el último juego y más novedoso que ha salido, lo primero que busca es cómo será la jugabilidad y lo segundo que busca siempre es los requisitos del juego para luego compararlos en los de su ordenador o portátil.

Aquí es lo novedoso en mi aplicación, toda esa tarea de buscar que juegos novedosos ha salido y la tarea de buscar los requisitos del juego, más si el ordenador será compatible o no lo brinda mi aplicación web y no he encontrado otra aplicación que realice específicamente dichas tareas.

2.2.Nombre y Logo

Mi aplicación tiene de nombre Virtual Gaming.

Es cierto que significado no explica mucho de qué va mi aplicación pero lo elegí porque hace referencia a una página muy usada en España que es Instant Gaming, una página que compra de juegos, diría que la parte de 'Virtual' da a entender que solo encontrará juegos en digital y no información de juegos en físico, no se dará información de tiendas físicas, y 'Gaming' solo informa sobre videojuegos, no de sus creadores ni la empresa.

Logo:



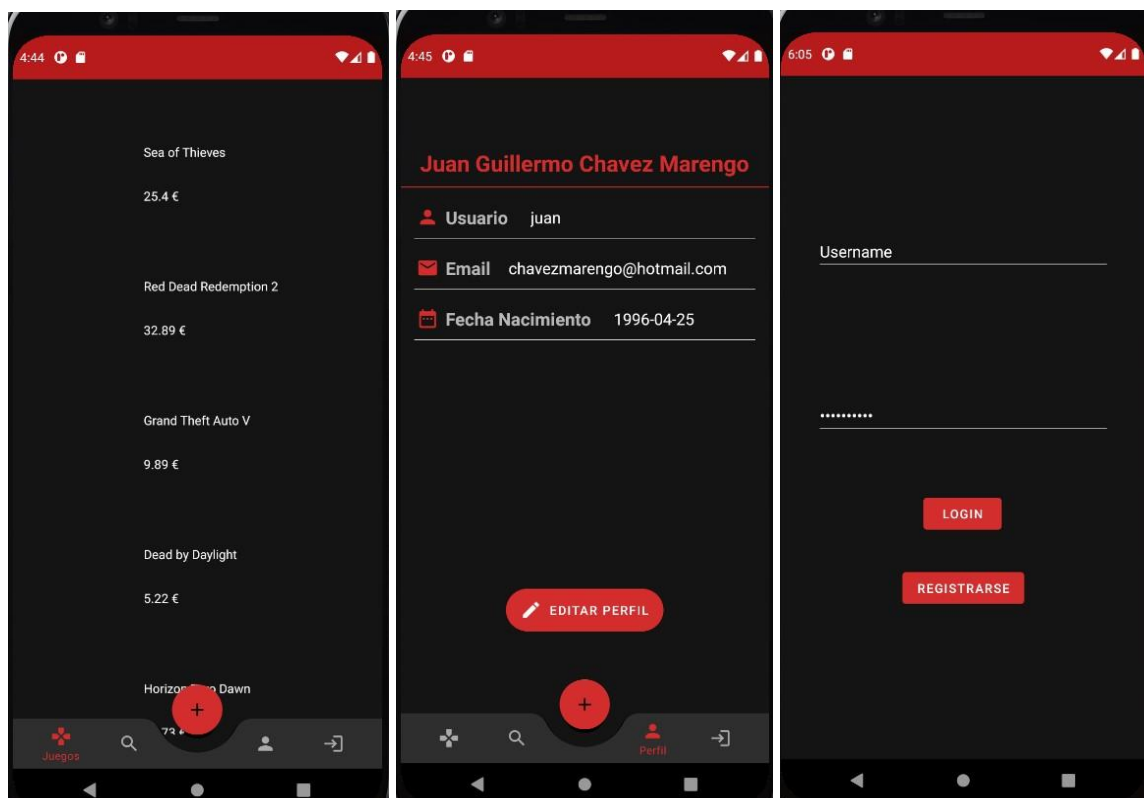
2.3.Producto

2.3.1.¿Qué vas a ofrecer?

Una aplicación web en android que informe sobre videojuegos y donde puedas agregar las especificaciones de tu ordenador y la aplicación se encargará de compararlos con los requisitos del juego y te dirá si está apto o no para jugar en tu ordenador o portátil.

Podrás crear una sesión y loguearte, para así tener tu lista de ordenadores y juegos favoritos a tu gusto.

Capturas: del listado y el perfil del usuario logueado












2.3.2.Finalidad

Da la facilidad al usuario de elegir un pc adecuado al momento de comprarlo. Cubre la necesidad de saber si el ordenador del usuario es apto para un juego en específico y si tiene que mejorarlo para jugarlo.

2.4. Modelo de Negocio

Lienzo de Modelo de Negocios

Socios Clave  <ul style="list-style-type: none"> -Socio con alguna pequeña empresa que venda videojuegos o componentes de pc. -Información sobre lo mas valorado en videojuegos y en componentes de pc. -Brindar informacion de ofertas de viduegos y lo ultimo que saldrá a la tienda. 	Actividades Clave  <ul style="list-style-type: none"> -Investigar para que le lleguen notificaciones al usuario sobre juegos. -En la tienda de la app los usuarios podran dejar su opinion y critica. -La app contará con un modo premiun. Recursos Clave  <ul style="list-style-type: none"> -Equipo de desarrollo -Servicios de consultaria por medio de las redes sociales -Asociación con otras paginas de videojuegos. 	Propuesta de Valor  <ul style="list-style-type: none"> -Los usuarios se aseguran antes de comprar un juego. -Ayudamos a los usuarios que tienen pensado comprarse un pc pero aun no se deciden cual. -Satisfacemos la necesidad de distraerse un rato y pasarla lo mas bien posible. 	Relación con Clientes  <ul style="list-style-type: none"> -En la parte de comentarios de la app el usuario podra evaluar la aplicacion. -Dejamos un telefono de soporte -Tenemos una red social al en el cual tambien podremos mantener una conversacion con el usuario. Canales  <ul style="list-style-type: none"> -Facebook -Twitter <p>En ambas redes sociales pueden informarse sobre nuevas actualizaciones de la app y las ofertas del modo premiun.</p>	Segmentos de Cliente  <ul style="list-style-type: none"> -Usuarios domesticos que usen nuevas tecnologías y residan en las grandes ciudades. -Los usuarios que recaudan dinero jugando videojuegos o influencers.
Estructura de Costos  <ul style="list-style-type: none"> -Costes de desarrollo -Cuotas mensuales a los encargados de las redes sociales -Costos de marketing 		Fuente de Ingresos  <ul style="list-style-type: none"> -Modo premiun de la aplicacion 		

2.5.Consumidores y Posibles Clientes

2.5.1.Tipo de consumidores

- Usuarios interesados en los videojuegos
- Usuarios que tengan pensado comprarse un pc

2.5.2.Zona en la que viven

Normalmente el mayor número de usuarios provendrá de los que vivan en las ciudades grandes ya que en algunos pueblos no tienen la posibilidad de encontrar tiendas físicas de componentes de pc o que los servicios de mensajerías no puedan llegar hasta sus hogares.

2.6.Competencias

Algunas app competentes que podría encontrar son las que informan sobre videojuegos recientes y dan una información más reciente y detallada del juego, más de lo que mi app ofrece y otras que informan sobre componentes de pcs y sus ofertas.

2.7.DAFO

2.7.1.Debilidades

A la aplicaciones le faltan muchas cosas que hoy en día en otras aplicaciones de juegos lo tienen más que pulido, como filtros de búsqueda.

Mi aplicación no posee mayor información de juegos ni direcciones de página para seguir informándote o comprar.

2.7.2.Amenazas

Podrían sacar a la tienda otra app parecida a la mía o hasta mejorada y mucho más completa.

La desactualización podría conllevar a las quejas por parte de los usuarios por lo que tendríamos que estar muy atento a lo último en novedades.

2.7.3.Fortalezas

No se necesita de mucho personal para mantener la aplicación en constante proceso.

No se gastará más dinero luego de haber concluido con las mejoras futuras a cumplir.

2.7.4.Oportunidades

Tiene muchas oportunidades en el campo tecnológico.

A nivel de gente influencer los ayuda a los que se dedican a crear contenido competitivo, a tener una pc adecuada para su trabajo.

2.8.Publicidad y Promoción

La publicidad la podríamos transmitir desde las redes sociales hasta con ayuda de algunos influencers dispuestos.

Cada cierto tiempo podría haber una muestra gratis de algunos días del modo premium de la aplicación, así donde a saber a los que aun no lo compran lo rentable que sería tenerlo comprado y tener al app al completo con todo y sus nuevas características.

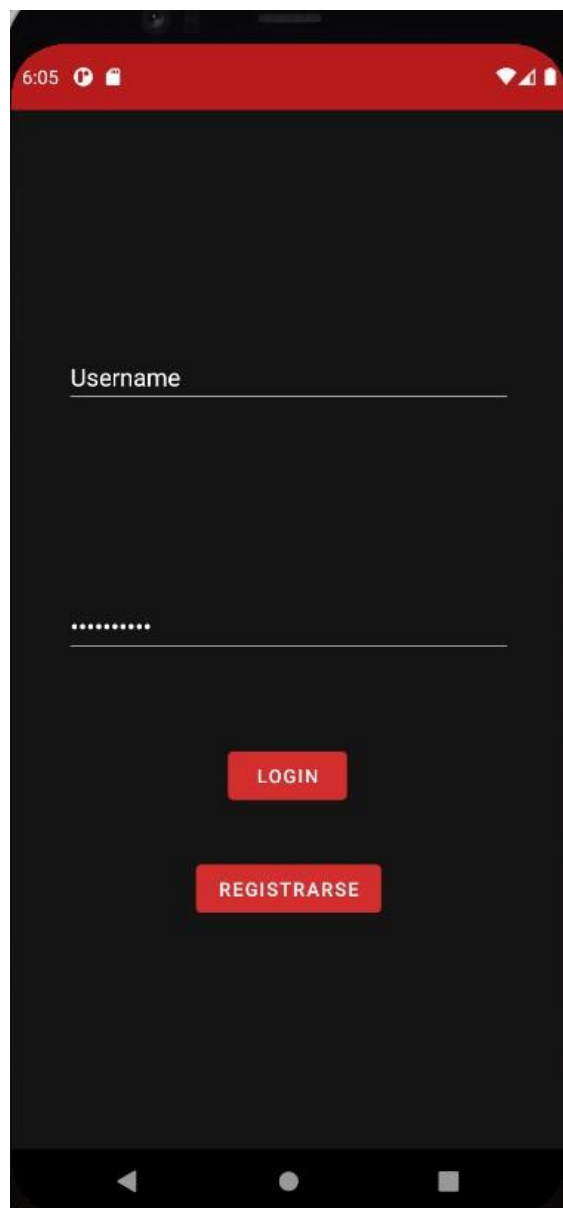
Se tendría que ver también hasta cuanto se puede invertir en la publicidad ya que el modo premium de la app a la vez que no se conseguirá mucho ya que no cuesta tanto y el pago a las personas encargadas de dar soporte y encargados de las redes sociales, el gasto podría ser mayor.

3.Descripción Detallada del Sistema

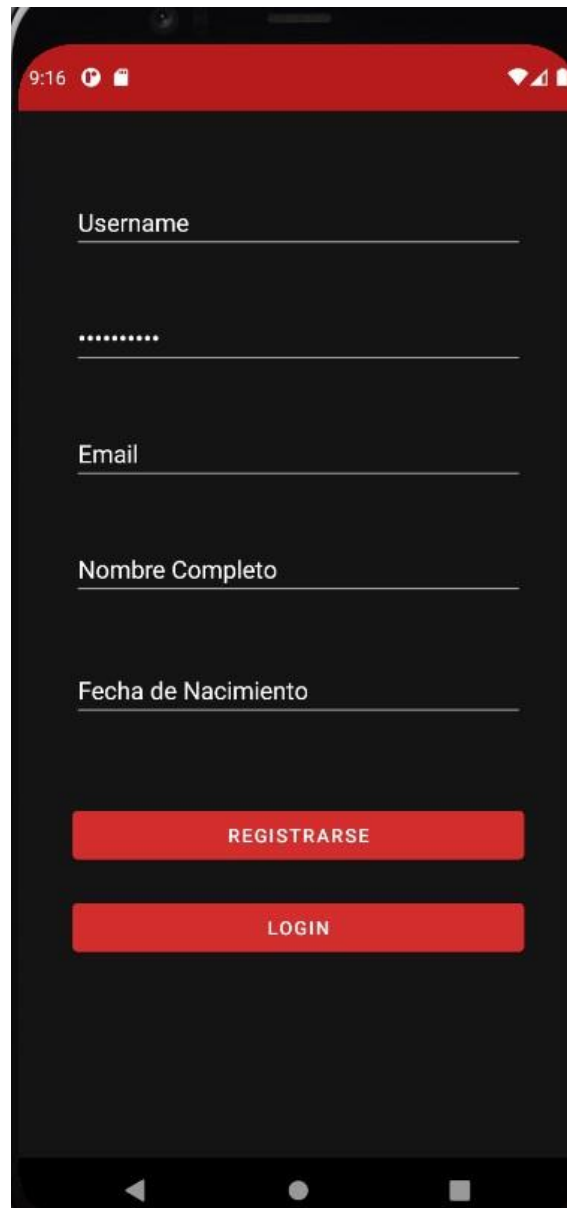
Al iniciarse la aplicación mostrará una pantalla de login con dos entradas de texto, uno para el nombre de usuario y el otro para la contraseña, y con dos botones, uno para loguearse y otro para registrarse.

Usuario:

El usuario al ser nuevo tendrá que darle al botón de registrarse para crear una nueva sesión.



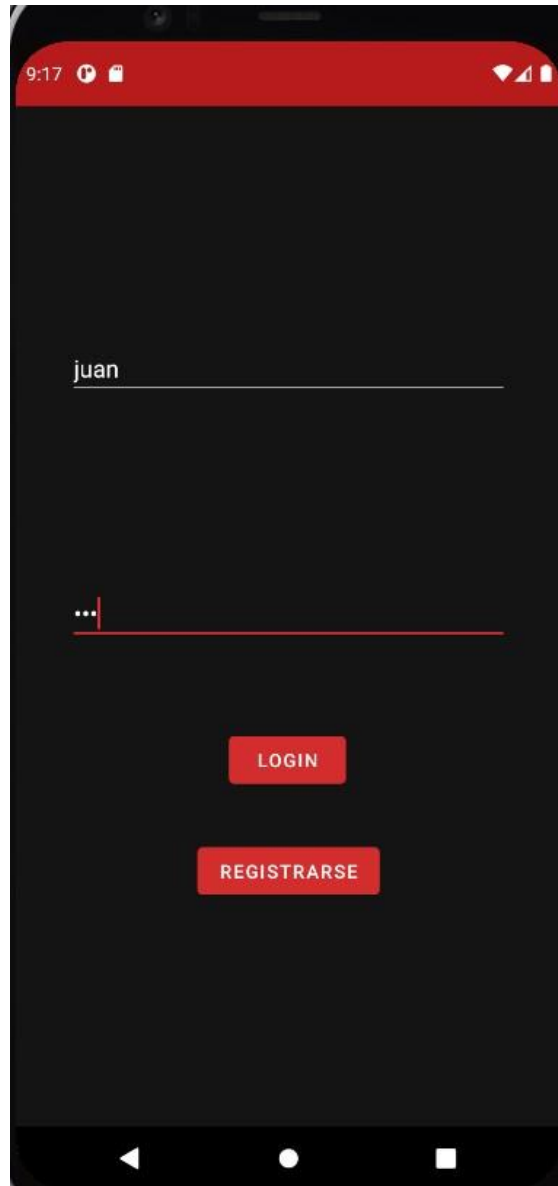
En la pantalla de registro pedirá el nombre completo, el nombre de usuario, el email, la fecha de nacimiento y la contraseña.



The image shows a mobile application interface for registration. It features a dark background with white text and red buttons. The form includes the following fields from top to bottom: a 'Username' field, a password field represented by a series of dots, an 'Email' field, a 'Nombre Completo' (Full Name) field, and a 'Fecha de Nacimiento' (Date of Birth) field. Below these fields are two prominent red buttons: 'REGISTRARSE' (Register) and 'LOGIN'. The top of the screen shows a red status bar with the time '9:16' and various icons. The bottom of the screen displays the standard Android navigation bar.

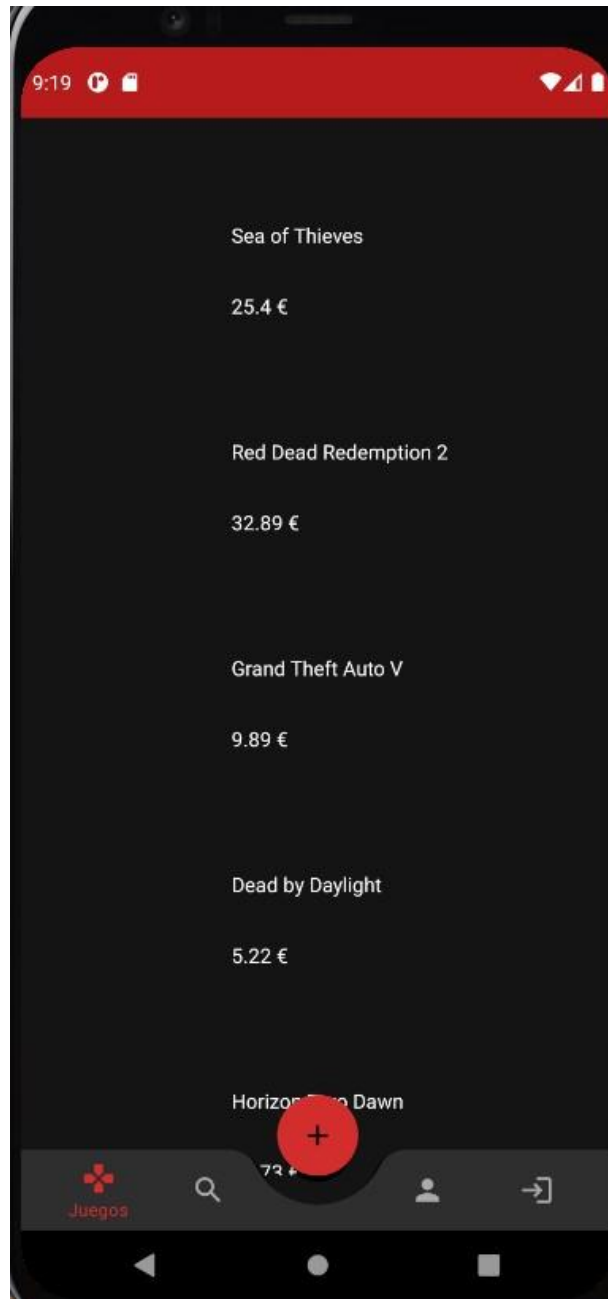
Luego de haberse registrado lo siguiente que puede hacer es dar click a registrarse y le permitirá volver a la pantalla de login solo si ha ingresado los datos correctamente y dejar ni uno vacío.

En la pantalla de login podrá usar el nombre de usuario y contraseña creado hace un momento para luego darle al botón de login.



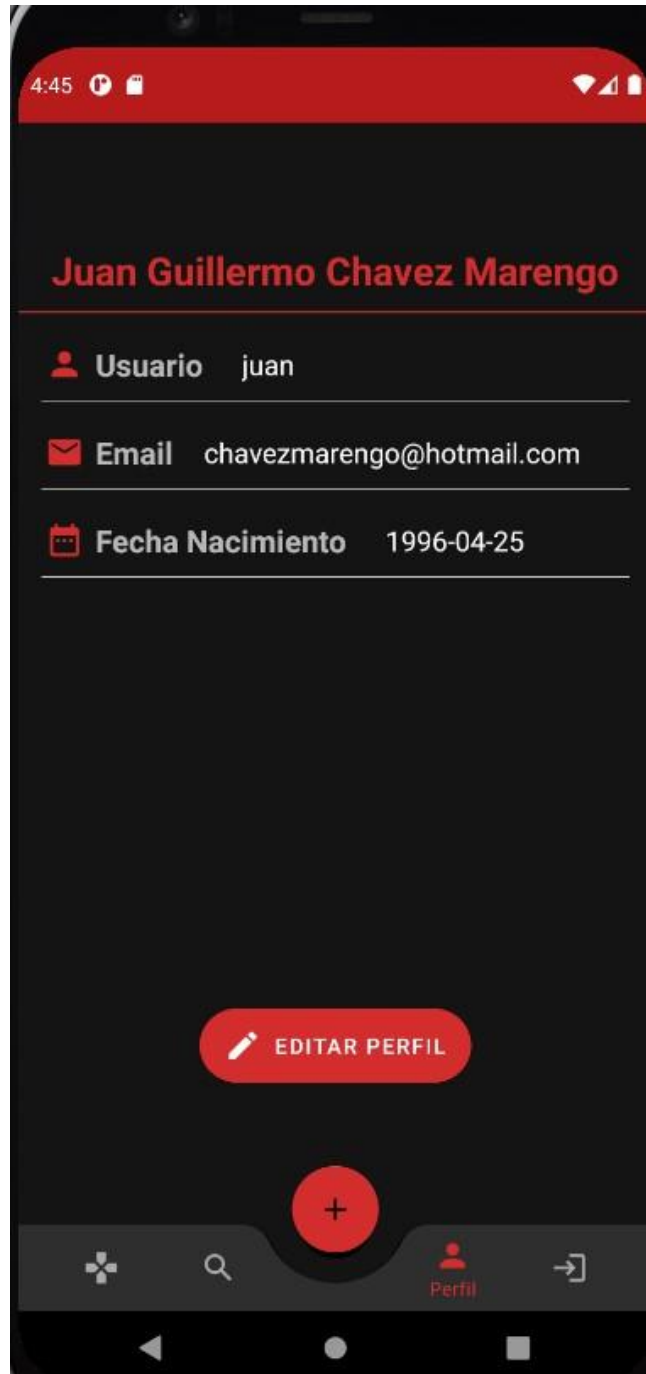
Una vez haber iniciado sesión se observará la pantalla principal del listado de videojuegos.

En el listado aparecerá el videojuego con una imagen del propio juego más el nombre y el precio.



También podrá acceder a su perfil dándole al icono de la persona en el menú inferior.

En la pantalla del perfil podrá visualizar los datos ingresados en el registro, como su nombre completo del usuario, el usuario, el email y la fecha de nacimiento ingresada.



Si le en la barra de menú le da al botón circular del medio lo enviará a otra ventana en donde le permitirá ingresar un ordenador nuevo.

Le pedirá un título para su ordenador y que elija entre los diferentes procesadores, memorias ram y gráficas que tenga su ordenador.

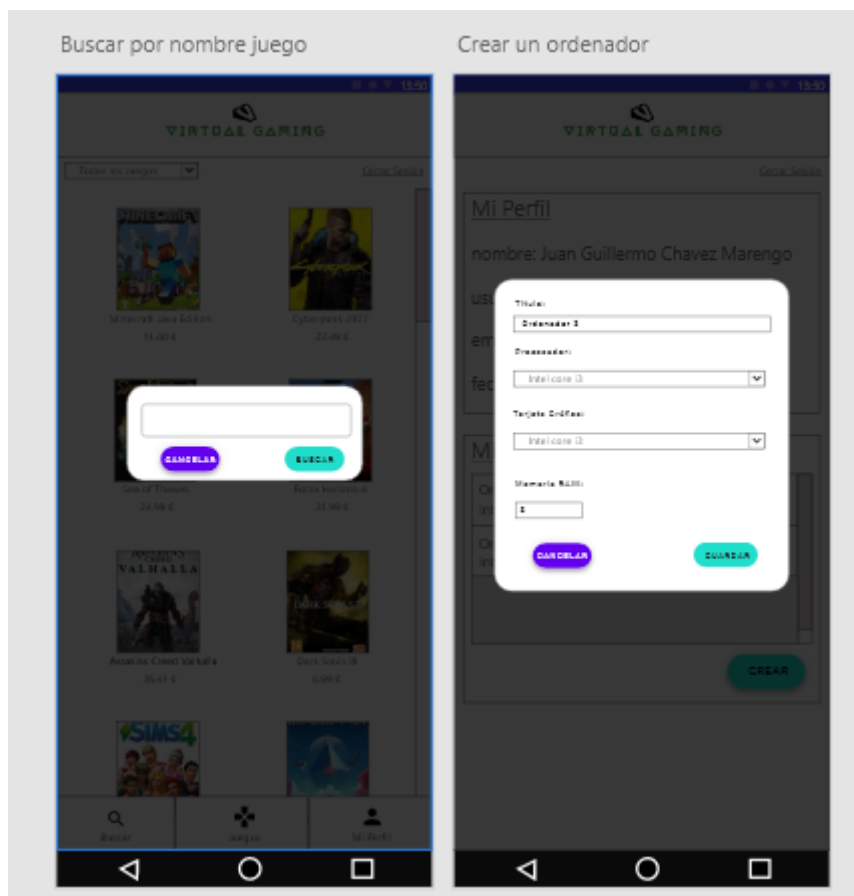
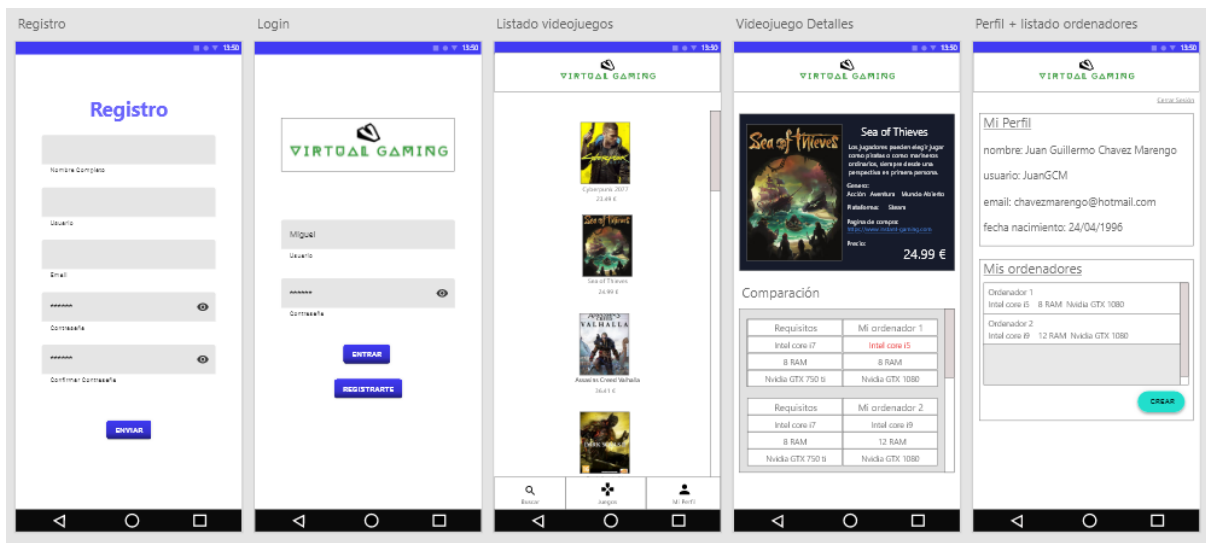


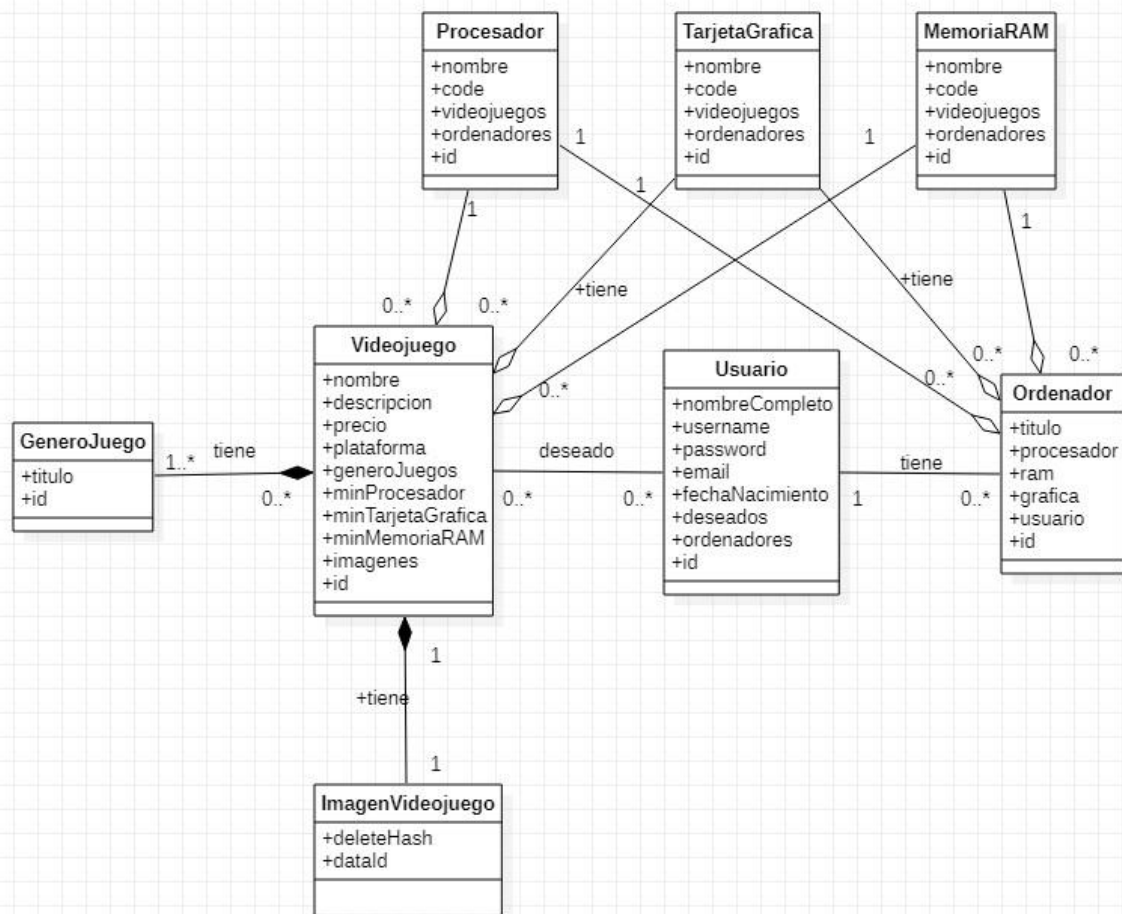
The screenshot shows a mobile application interface with a dark theme. At the top, a red status bar displays the time 6:39 and various icons. Below the status bar, the app title "Virtual-Gaming" is visible. The main content area contains a form with the following fields:

- Titulo:** A text input field containing "ordenador 1".
- Procesador:** A dropdown menu showing "Intel core 2 duo".
- Memoria RAM:** A dropdown menu showing "6GB DDR2".
- Tarjetas Graficas:** A dropdown menu showing "NVIDIA GTX 750 ti".

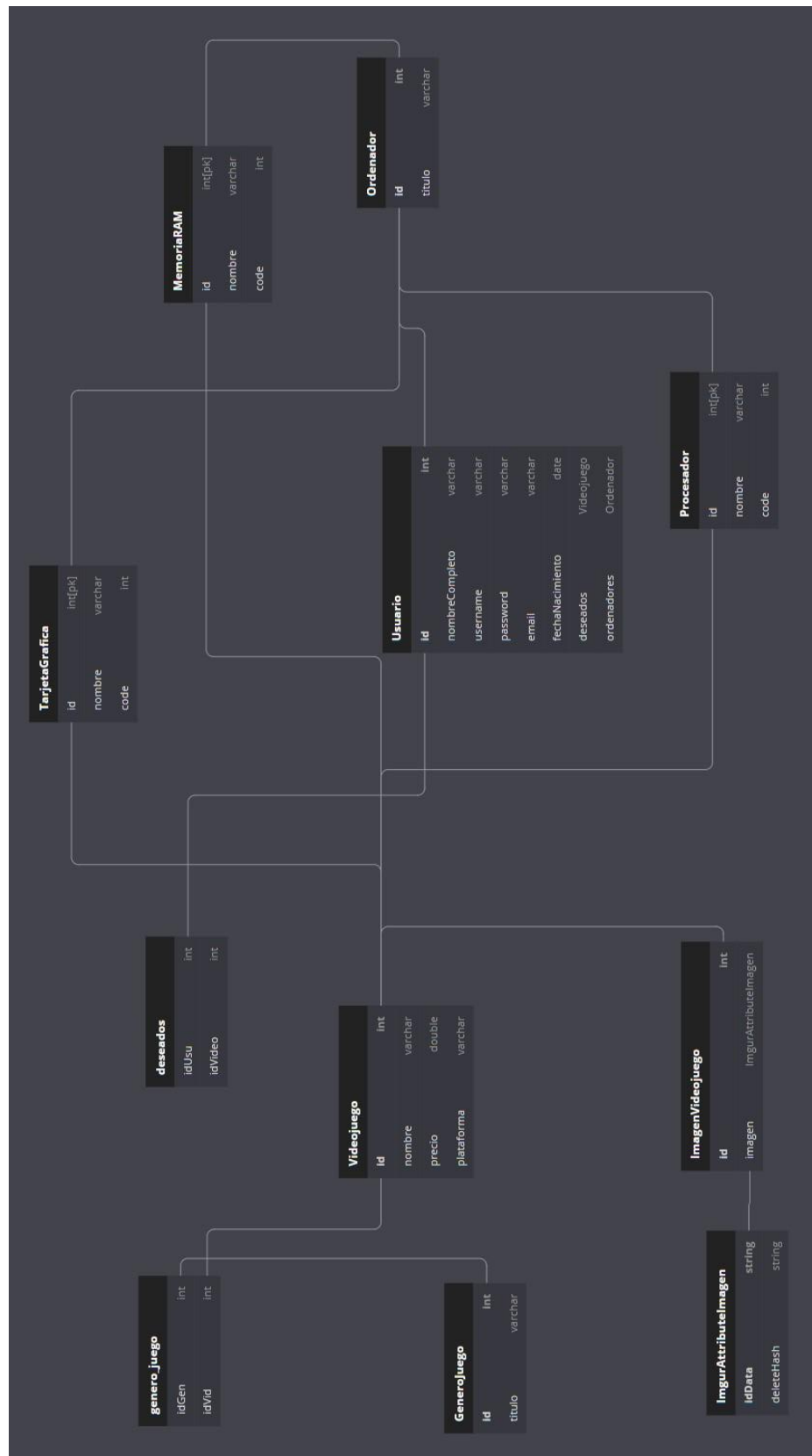
At the bottom of the form is a red button with the text "GUARDAR". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

4.Modelado





5.Diseño



6.Implementación

6.1.Paquetes y Clases de la API

Paquete Controllers: paquete que contendrá los controladores que realizarán las consultas a la base de datos.

- Clase GeneroJuegoController: hace consultas un listado de géneros de juegos y para crear y eliminar un género.
- Clase ImagenController: se encuentran las consultas para obtener un listado de imágenes, conseguir uno en específico, para crear y eliminar.
- Clase MemoriaRAMController: hace consultas a la ddbb para conseguir el listado de rams y para crear y eliminar una.
- Clase ProcesadorController: hace consultas de listado, crear y eliminar un procesador.
- Clase TarjetaGraficaController: hace las consultas necesarias para el crud de tarjeta grafica mas su listado.
- Clase RestController: gestiona las incidencias del proyecto
- Clase UsuarioController: realiza las peticiones de crear un usuario nuevo y obtener un usuario por el username
- Clase VideojuegoController: hace peticiones para listado, crear juego nuevo, buscar juego por plataforma, modificar un juego, conseguir un juego por id y eliminar un juego.
- Clase OrdenadorController: se encarga de hacer las peticiones de crear, listado y eliminar ordenador.

Paquete DTOS: contiene los dtos los cuales puede contener información de múltiples tablas y concentrarlas en una.

- VideojuegoDTO: crea dos dtos de Videojuego, en uno pide el los datos a presentar en el listado y el segundo dto es para mostrar los detalles del videojuego.
- GeneroJuegoDTO: crea un dto de género juego para solo leer el título e ir del género.
- ImagenDTO: crea dos dto en donde uno llama a una variable videojuego y un texto, y el otro llama al dataId de la imagen mas el id de la entidad imagen.
- MemoriaRAMDTO: crea un dto para memoriaram en donde usa el título y el code de la memoria.
- OrdenadorDTO: crea un dto para Ordenador en donde llama al título, al procesador, ram, y a la gráfica.
- ProcesadorDTO: crea un dto de Procesador usando el id, titulo y code el mismo.
- TarjetaGraficaDTO: crea un dto de la clase Tarjeta Gráfica usando el id, título y el code.
- UsuarioDTO: crea un dto para la clase Usuario y llama al nombre, username, email, fechanacimiento, e id.

Paquete Errors: contiene las Clases que controlaras los errores y excepciones dadas por el programa al ejecutarse.

- Errores: contiene la clase ApiError que con el especificamos los httpstatus que nos da el sistema al hacer una petición , también contiene la clase ApiValidationSubError para configurar los mensajes que saldrán al momento de fallar una validación.

- Excepciones: contiene a la clase `SingleEntityNotFound` para controlar la excepción que salta cuando no encuentra a la entidad, también contiene a las clases que controlan cuando no encuentra a una entidad cuando la buscan por su id, cuando buscan elementos de un tipo específico en una consulta y hasta cuando no encuentra una imagen.

Paquete `Imageupload`: contiene a las varias clases que serán las encargadas del correcto funcionamiento de las solicitudes a la api de imgur.

- `BasicImageStorageService`: es un servicio vital para el almacenamiento de imágenes, el cargar imágenes de nuestro recurso o eliminar uno.
- `ImgurFilesController`: es un controlador que hace la consulta de conseguir un archivo de la api en imgur.
- `ImgurImageAttribute`: tiene la propiedad `id` de la imagen que se necesita para cargar la misma y el `deletehash` para la eliminación de la imagen.
- `ImgurImageAttributeToStringConverter`: se encarga de traducir toda la información necesaria desde y hacia un string
- `ImgurService`: es un servicio en donde más que todo se declaran constantes para las consultas hasta la api de imgur, también se encuentran las funciones de cargar imagen , conseguir uno específica y eliminar.
- `ImgurStorageService`: esta clase es el corazón de las subida de ficheros y ofrece métodos de almacenamiento de imagen y el delete para eliminarlo.
- `ImgurStorageServiceImpl`: esta clase implemente el ya explicado servicio de imgur para completarlos con mensajes y controlar algunos posibles errores.
- `MediaTypeUrlResource`: una clase la propiedad tipo de media y la url.

Paquete `Models`: contiene las distintas entidades del proyecto.

- Clase `Usuario`: contiene la entidad usuario el cual será usado para la autenticación del usuario.

- Clase Ordenador: es la entidad Ordenador que el usuario podrá crear al momento de entrar a la aplicación.
- Clase Videojuego: la entidad videojuego es lo que usará la aplicación para mostrar un listado y la información de los juegos.
- Clase GeneroJuego: la entidad género de juego sirve para describir al videojuego si es de acción o es un juego de aventuras.
- Clase ImagenVideojuego: define la entidad imagen que será usado como un elemento más del juego a mostrar.
- Clase Procesador: es la entidad que definirá qué procesador tiene un ordenador y como requisito para un juego.
- Clase MemoriaRAM: es la entidad que definirá qué memorias ram tiene un ordenador y como requisito para un juego.
- Clase TarjetaGrafica: es la entidad que definirá qué tarjeta gráfica tiene un ordenador y como requisito para un juego.

Paquete Repositories: contiene los repositorios de cada una de las entidades en donde se declararán algunas funciones a usar por los servicios.

- GeneroJuegoRepository: declara la función buscar por título de generojuego
- ImagenRepository: declara una funcion que busca imágenes por cada videojuego específico que encuentre.
- MemoriaRAMRepository: declara una función buscar memoria por videojuego y un buscar por código.
- OrdenadorRepository: tiene la función buscar por título y devuelve un listado de ordenadores
- ProcesadorRepository: define una función de buscar videojuego por procesador
- TarjetaGraficaRepository: encargado de definir la función de buscar videojuego por procesador

- **UsuarioRepository:** define la función `findByUsername` para buscar por nombre de usuario y la función para buscar por email.
- **VideojuegoRepository:** define las funciones necesarias para buscar un videojuego por precio, plataforma, nombre y likes.

Paquete Security: contiene a las clases que se encargaran de la autenticación del usuario y de crear el token para luego permitir al usuario realizar consultas a la base de datos.

- **AuthenticationController:** controlador que se encarga de las operaciones de autenticación y refresco del token.
- **ConfigurePasswordEncoder:** una configuración útil para codificar la password ingresada por el usuario.
- **JwtAuthenticationEntryPoint:** Componente que indica cómo comenzar el esquema de autenticación y define el reino de seguridad.
- **JwtAuthorizationFilter:** es un componente que extrae del token el usuario logueado y también tiene el servicio que proporciona un método que se encarga de entrar el token bearer de un encabezado de Authentication.
- **JwtTokenProvider:** crea las constantes para la seguridad y además se encarga de parsear tokens y de generar el token tanto para un usuario determinado o del que está actualmente autenticado.
- **WebSecurityConfiguration:** se encarga de dar permisos de uso de las consultas dependiendo el rol que tenga el usuario.

Paquete Services: contiene a los servicios del programa que nos permiten realizar el crud para cada entidad.

- **GeneroJuegoServicio:** define las funciones de crear un género de juego nuevo, mostrar un listado de géneros y eliminar por id.
- **ImagenServicio:** es un servicio de imagen que se encarga de guardar las imágenes subidas y de eliminarlas.

- MemoriaRAMService: es un servicio de la entidad memoriaram que se encarga de crear una memoria ram nueva, mostrar un listado de memorias y eliminarlo por id.
- OrdenadorServicio: es un servicio de ordenador que se encarga de crear un ordenador nuevo,
- ProcesadorServicio: es un servicio de procesador que se encarga de mostrar un listado de procesadores, crear uno nuevo y eliminarlo por id.
- ServicioBase: es un servicio que se encarga de definir las funciones básicas de guardar, mostrar un listado, eliminar, etc.
- TarjetaGraficaServicio: es un servicio de tarjeta grafica que se encarga de mostrar un listado de gráficas, crear uno nuevo y de eliminar un específico.
- UserDetailsServiceImpl: es un servicio que se encarga de cargar un usuario por el username y mostrar sus detalles.
- UsuarioServicio: es un servicio de usuario que se encarga de crear un usuario nuevo y buscarlo por username.
- VideojuegoServicio: es un servicio de imagen que se encarga de mostrar un listado de videojuegos, crear un juego nuevo, editar y eliminarlo, buscar por id

6.2.Paquetes y Clases de Android

Paquete Login: contiene las clases que hacen posible el login en la aplicación de android.

- LoginActivity: se encarga de recibir los datos ingresado por el usuario y aplicarlo en la consulta para la base de datos, en caso de ser nuevo lo envía a la vista registro.
- LoginRequest: es la clase definida para guardar los datos que el usuario ingrese al momento del login.
- LoginResponse: es la clase que guardará el token generado al momento de login.

- LoginService: es el servicio que consultará a la base de datos.
- User: es la clase encargada de guardar los datos devueltos por la base de datos.

Paquete Profile: contiene las clases que permiten mostrar en el emulador el perfil del usuario logueado.

- ProfileFragment: esta clase se encarga de mostrar en la vista de perfil los datos del usuario logueado
- ProfileViewModel: se encarga de enviar la consulta a la base de datos.

Paquete Registro: contiene las clases que permiten al usuario registrarse si es que aún no tiene un usuario creado.

- LoggedInUserView: es la clase que guardará el user name
- LoginFormState: es la clase que guardará los errores que pueda dar al momento del login
- LoginResult: es la clase que guardará el resultado del login, si es que fue satisfactorio o si hubo algún error.
- RegisterActivity: hace la llamada al método del servicio para registrarse.
- RegisterRequest: es la clase que guardará los datos al momento que el usuario se registre.
- RegisterResponse: es la clase que guardará los datos devueltos por la base de datos luego de registrarse el usuario.
-
- RegisterService: es el servicio que se encarga de hacer las consultas a la base de datos.

Paquete VideojuegoDetails: contiene las clases que permitirán mostrar los detalles de un juego en una ventana a parte.

- JuegoDetallesFragment: se encarga de mostrar en la vista de detalles los datos importantes del videojuego.

- MiObservable: encargado de enviar datos entre fragments.

Paquete Videojuegos: contiene las clases que muestran de primeras el listado de videojuegos.

- MyVideojuegoRecyclerViewAdapter: encargado de llenar los textview de la vista listado de videojuegos
- VideojuegoServicio: se encarga de hacer las consultas a la base de datos.
- VideojuegosFragment: se encarga de enviar los datos devueltos por la base de datos al RecyclerViewAdapter
- VideojuegosViewModel: se encarga de iniciar el retrofit y llamar a la función que luego llama al servicio que iniciará la consulta a la base de datos.

Paquete Models: contienen los data classes que lo usaremos para recibir datos de la base de datos y mostrarlos en las vistas.

- UnVideojuego: es la clase definida para recibir los datos de videojuego devuelto por la base de datos.
- UnJuegoDetalle: será la encargada de contener todos los datos del videojuego, incluyendo la imagen como sus componentes.
- Procesador: define un procesador y sus atributos
- MemoriaRAM: define una memoria ram y sus atributos
- TarjetaGrafica: define una tarjeta grafica y sus atributos