



PROGRAMACIÓN DE SISTEMA 22/23 Q1



Aplicación de Búsqueda de Eventos

EventsTime

Autores: Julián Villamor Barreiro
Mario López Cea
Diego Veiga Couceiro
Fecha: *A Coruña, 15 Octubre 2022*

Índice

Capítulos	Página
1. Introducción	1
1.1. Objetivos	1
1.2. Motivación	1
1.3. Trabajo relacionado	1
2. Análisis de requisitos	1
2.1. Funcionalidades	1
2.2. Prioridades	2
3. Planificación inicial	3
3.1. Iteraciones	3
3.2. Responsabilidades	4
3.3. Hitos	4
3.4. Incidencias	5
4. Diseño	5
4.1. Arquitectura	5
4.2. Persistencia de datos	6
4.3. Vista	7
4.4. Comunicaciones	7
4.5. Sensores	7
4.6. Trabajo en segundo plano	7
5. Diseño IT-1 e IT-2	8
5.1. Datos	8
5.2. Dominio	8
5.3. Actividades	8
6. Control de versiones	8

Cuadro 1: Tabla de versiones.

Versión	Fecha	Autor
1.0	15/10/2022	Julián, Mario, Diego
1.1	09/11/2022	Julián, Mario, Diego
x	y	

1. Introducción

1.1. Objetivos

El objetivo principal de la aplicación es la creación de planes para organizar eventos con gente desconocida. Esto permitiría a las personas poder quedar con otras que tengan intereses en común y que de otra forma no podrían quedar con ellas al no conocerlas.

1.2. Motivación

La motivación por la cual hemos decidido crear esta aplicación fue la de intentar ayudar a las personas que no tengan a gente conocida para poder realizar actividades, por ejemplo, porque están en una ciudad nueva o porque en su círculo de amigos no haya nadie que tenga una determinada afición.

1.3. Trabajo relacionado

La aplicación meetup es una aplicación ya consolidada en el mercado con más de 10 millones de descargas, su objetivo es el mismo que el de nuestro proyecto, crear actividades y poder unirse a ellas. A diferencia de nuestra app esta ya tiene una base sólida con más funcionalidades como chat, permitir cobrar entra o crear grupos, y una base de usuarios fieles, sin embargo, nosotros nos centraremos en crear una experiencia sencilla y sólida centrará en el manejo de las actividades y si el usuario quiere chatear o realizar otras tareas delegarán en otras app.

2. Análisis de requisitos

2.1. Funcionalidades

Las prioridades se definirán en función de su posición en la lista cuanto mas arriba mas prioritario. Las funcionalidades principales de la aplicación como administrador serán:

1. Crear un plan/evento
2. Eliminar plan/evento

3. Modificar un plan ya creado

4. Editar participantes

Las funcionalidades principales como participante serán:

1. Registrarse en la app

2. Iniciar sesión en la cuenta

3. Eliminar cuenta

4. Buscar actividades

5. Búsqueda de un usuario

6. Editar datos de cuenta

7. Búsqueda filtrada de las actividades

8. Solicitar asistencia a una actividad

9. Eliminar la asistencia a una actividad

10. Puntuar usuario con el que compartiste actividad

2.2. Prioridades

De todas las funcionalidades definidas habrá unas mas importantes, que denominaremos core, y otras que serán funcionalidades adicionales que mejorarán la experiencia de usuario, pero que no son imprescindibles.

Las funcionalidades que consideramos que son imprescindibles para el funcionamiento de nuestra app serían: Registrarse en la app e iniciar sesión, funcionalidades básicas para poder empezar a utilizar la aplicación, a partir de estas tendríamos: Crear un plan/evento, modificar y eliminar un plan/evento y editar participantes, en lo que sería la administración de un evento; y para el usuario tendríamos: editar datos de cuenta, eliminar cuenta, buscar actividades, solicitar asistencia a una actividad y eliminar la asistencia a una actividad.

Por otro lado, otras funcionalidades no tan importantes, pero que si que pueden mejorar la experiencia para el usuario serían; búsqueda filtrada de las actividades, puntuar al usuario con el que compartiste una actividad y búsqueda de un usuario.

Entre las funcionalidades que se han definido existen una serie de dependencias que hay que definir para un correcto desarrollo de la aplicación.

La principal dependencia entre las funcionalidades es que para poder usar la aplicación, y por lo tanto hacer uso del resto de funcionalidades, es imprescindible estar registrado. Por lo tanto para ejecutar cualquier funcionalidad de la aplicación se necesitará registrarse y posteriormente iniciar sesión. Todas las funcionalidades que

dependen de estar registrado y con sesión iniciada, es decir, aquellas funcionalidades que un usuario puede ejecutar tras iniciar sesión son: editar datos de la cuenta, eliminar cuenta, buscar actividades, hacer una búsqueda filtrada, solicitar asistencia a una actividad, búsqueda de un usuario y crear un evento.

Además existen otras dependencias con estas funcionalidades. Para poder modificar o eliminar un plan es necesario que primero el plan se haya creado. Para poder modificar los participantes de un plan/evento tiene que haber participantes ya adscritos al evento. Para poder eliminar la asistencia a una actividad, el usuario debe de haber solicitado la asistencia a la actividad. Para puntuar al administrador de una actividad o a los participantes, el usuario debe de estar apuntado en esa actividad y además solo se podrá puntuar una vez la actividad haya terminado.

3. Planificación inicial

3.1. Iteraciones

IT-1 - Persistencia de Datos

En un primer momento tras haber decidido el modelo de datos y que base de datos utilizar, se creará una capa modelo y una capa de conexión de la base de datos por si se deciden realizar cambios en esta en el futuro no afectar al resto de aplicación. Se definirán las operaciones que queremos realizar a la base de datos y las interfaces para poder acceder a ellas desde la vista, posteriormente se crearán test automatizados para comprobar su funcionamiento.

En paralelo se trabajará en una vista básica para la disposición de las actividades haciendo uso de unos datos precreados.

IT-2 - Autenticación y cuenta

Una vez hecho todo el trabajo de abstracción y definición de interfaces para acceder a la db, se remplazará en la vista creada los datos pregenerados por la correspondiente función para la obtención de estos. También se definirán las vistas para la autenticación, iniciar sesión y registrarse, e implementar un sistema de persistencia para que el usuario no tenga que iniciar sesión cada vez que abra la app. Con esto se creará la vista de perfil que también nos ayudará a comprobar que el usuario se crea bien y se obtiene su información correctamente.

IT-3 - Actividades

En esta iteración nos centraremos en la gestión de las actividades, añadiendo las funcionalidades para crearlas, modificarlas, gestionarlas, unirse y salirse de ellas. También se añadirá la vista de la actividad que mostrará toda su información a los usuarios interesados en ellas.

IT-4 - Búsqueda y Filtrado

Una vez tengamos todo el sistema de las actividades montado y funcionando, añadiremos una funcionalidad clave para el usuario, la búsqueda y filtrado de datos. Para esto crearemos los respectivos elementos visuales que nos permitan realizar esta ac-

tividad, y mediante las interfaces creadas previamente obtendremos los datos de forma selectiva en función de lo que el usuario decida.

IT-5 - Social

Después de implementar todo lo principal, implementaremos lo relacionado con los usuarios, puntuar usuarios y crear un perfil visible para que el resto pueda verlo, también se añadirá funcionalidades para modificar el perfil del usuario.

IT-6 - Vista

Una vez implementadas todas la funcionalidades se pulirá y mejorará si hiciese falta la UI/UX de la aplicación para hacer que se sienta cómoda,sencilla y rápida para la persona que la usa.

3.2. Responsabilidades

La repartición en el primer hito fue la siguiente :

Julián

1. Capa de Datos
2. Casos de uso del primer hito
3. ViewModel Actividad main
4. Actividad info del usuario

Mario

1. Actividad Inicio Sesión
2. Actividad Registro usuario

Diego

1. ViewModel ActivityListPres
2. ViewModel ListActivities
3. ViewModel ListActivitiesAdapter
4. View ActivityListActivities

3.3. Hitos

Hito 1

Este hito englobara las 2 primeras iteraciones

Hito 2

Se incluirá en la app la IT-3 ,IT-4 y IT-5

Hito 3

Iteración final que contendrá IT-6

3.4. Incidencias

Antes de avanzar una iteración se ha de probar que la anterior funciona correctamente si no estaremos acarreando problemas y al final habrá mucho trabajo que hacer, hay que tener especial cuidado en la primera iteración ya que un modelo mal hecho implicara que la mayoría de funcionalidades que dependen del tengan un mal funcionamiento.

4. Diseño

4.1. Arquitectura

mvvm(Model-View-ViewModel) nos permite dividira la aplicacion en capas, la primera el modelo para representar la parte de datos, la Vista y el ViewModel que conecta la vista con el modelo y notificar a la vista si han sucedido cambios.

En combinacion con el mvvm para una mejor escalabilidad y futuro mantenimiento también se implementara **Clearn Architecture**, en nuestro caso la dividimos en 4 capas :

- La capa core definen funcionalidades comunes para todo el proyecto, como la inicialización de la db.
- Para la persistencia se definió una Capa de datos que a su vez consta de 3 subapartados :
 - **Model** : aquí se definen las clases que representan a los objetos, en nuestro caso el usuario y la actividad.
 - **Network** : En este apartado definimos las clases que interactuaran con la base de datos, el resultado será igual aun que se cambie la db, para eso se usan interfaces. Estas clases se nombraran como *service*
 - **Repository** : Es la capa que expone las funcionalidades al dominio, aquí se decide que ser *service* usar, se pueden combinar múltiples servicios.
- La capa dominio implementara los casos de uso haciendo uso del repository para interactuar con la db.
- La capa de la vista donde tendremos las **Actividades** y los **ViewModels**, los intermediarios para interactuar con el Dominio y obtener los datos.

4.2. Persistencia de datos

Los datos con los que se trabajará los vamos a dividir en dos tipos, los datos de una actividad y los datos de los usuarios, para su persistencia se decidió el uso de Firebase para el almacenado y la autenticación.

Los datos con los que trabajara cada actividad serán:

- ID
- Título
- Descripción
- Fecha y hora de inicio
- Fecha y hora de fin
- Fecha y hora de creación
- Lugar, Coordenadas
- Administrador de la actividad
- Participantes
- Categorías

Los datos con los que trabajará cada usuario serán:

- ID
- Nombre
- Apellidos
- Fecha de nacimiento
- Imagen de perfil
- Descripción
- E-mail
- Contraseña
- Teléfono
- Link redes sociales (instagram, twitter, facebook)
- Puntuación media
- Intereses

4.3. Vista

Actividades

- Actividades/planes
- Creación Actividad
- Modificación Actividad
- Perfil Usuario
- Modificar Perfil
- Inicio de Sesión
- Registro de Usuario
- Puntuación Usuarios
- Búsqueda Usuarios

Fragmentos:

No necesarios en principio

Navegación:

Se dispondrá de una barra de navegación donde el usuario podrá navegar entre las diferentes actividades, también se intentara implementar que se pueda cambiar haciendo *swipe*, para los filtros se lanzara un menú donde se seleccionara cuales desea aplicar.

4.4. Comunicaciones

Se establecerá una capa que nos permitirá interactuar con el servidor desde el modelo de forma agnóstica, es decir que se proporcionarán una interfaz que implementara unas funcionalidades base para trabajar con ella y si en algún momento se cambia no afectará al resto de la aplicación, la interfaz será la misma solo cambiaremos la implementación.

4.5. Sensores

No hace falta sensores

4.6. Trabajo en segundo plano

Prefetchig en segundo plano de actividades en cache para experiencia mas ágil

5. Diseño IT-1 e IT-2

5.1. Datos

En estas primeras iteraciones se implemento toda las funcionalidades core en la capa de datos, primero definimos los modelos de user y actividad para obtener y almacenar los datos de forma consistente, despues creamos los servicios para el user las actividades y la autenticación. Y para acabar definimos el repositorio haciendo uso de los servicios.

5.2. Dominio

En esta capa solo se implementaron los casos de uso que requerían las vistas, en proximas iteraciones se iran implementando el resto progresivamente.

5.3. Actividades

Para esta primera iteración definimos 5 actividades/Vistas :

- **Main** Esta actividad es el punto de partida de la app, dependiendo si la sesión esta iniciada o no nos llevara o bien a la actividad de login y register o si no nos mostrara una lista de botones para probar las funcionalidades implementadas.
- **Login** Esta actividad nos permitirá iniciar sesión, si se hace con las credenciales correctas volveremos al main si no saldrá un aviso para indicar que algo salio mal
- **Register** Actividad para registrar un nuevo usuario, al igual que antes si se hace de manera correcta iremos al main y si no saldrá un toast, en esta iteración no se podrá subir la imagen de usuario.
- **User Info** Actividad simple para ver los principales datos del usuario actual
- **Activity List** Actividad que nos mostrara mediante tarjetas en un recycler-View las ultimas actividades y su información.

6. Control de versiones

https://github.com/xiao-villamor/TT_PS.git