

## **Proyecto: Aplicación móvil para reseña de artículos**

### **Contexto**

Su equipo ha sido contratado para la creación de una aplicación en Android a través de la cual los usuarios puedan reseñar y compartir sus opiniones acerca de <Artículo>. Encontramos actualmente varias de estas aplicaciones como lo son goodreads para la reseña de libros y letterboxd para películas. El cliente espera que a través de la aplicación se pueda crear una pequeña red social, donde los usuarios puedan calificar y comentar <Artículo> y al mismo tiempo formar comunidad con funcionalidades básicas, como seguir otras cuentas y poder estar actualizado acerca de sus opiniones.

### **¿Cómo se imagina la aplicación?**

Respuesta de cliente: “Quiero una red social de <Artículo>. Lo más básico que me imagino es que sea posible crear una cuenta e ingresar únicamente a todas las funcionalidades si se está autenticado. Los datos mínimos que necesito de todo usuario es su correo, un nombre de usuario, y la contraseña. Si quieres recopilar datos adicionales podría ser un plus, pero es importante que primero lo discutamos.”

“Ahora respecto a las pantallas. Te voy a decir las principales que me imagino, sin embargo, si se te ocurre alguna adicional lo hablamos. La más importante es una donde pueda ver los reviews o análisis que han realizado las personas que sigue el usuario, obviamente siempre priorizando las últimas actualizaciones. Por otro lado, tiene que haber una pantalla donde se muestren los últimos <Artículos>. Sería bueno que se pudiera realizar algún tipo de filtro por subcategoría para que solo se muestren los artículos de interés para el usuario. Creo que una barra de búsqueda también podría ser de utilidad para buscar fácilmente por nombre.”

“Pasemos a lo principal: cuando el usuario selecciona un <Artículo> puede leer algunos datos principales de este y podrá leer las reseñas del resto de usuarios. Pero lo más importante es que pueda calificar el artículo y realizar un review/análisis de este. La calificación se realizará entre 0 y 5 usando únicamente números enteros. Tener cuidado de que el usuario únicamente pueda realizar una calificación por artículo, en caso de ser necesario se puede modificar la calificación y análisis inicial.”

“Dado que quiero que esto se comporte como una red social, es importante que exista un sistema para realizar comentarios a los análisis. Lo ideal es que cualquiera pueda realizar comentarios a otro usuario así no lo siga. ¿Qué piensas de generar

un mecanismo de likes para los reviews o comentarios? ¿Sería muy complicado? Bueno después lo podemos discutir”

“Ahora pasemos a las cosas menores. Quiero una pantalla donde el usuario pueda modificar su perfil. Por ejemplo, es importante que el usuario pueda poner una foto para ser reconocido dentro de la comunidad. También recuerda poner la opción de eliminar la cuenta. Ahora dentro del perfil (u otra pantalla) quiero que el usuario pueda ver todos los <Artículo> que ha calificado, para que pueda tener seguimiento de lo que más le gusta.”

“Por último, quisiera que haya notificaciones, estas pueden ser cuando ocurra un evento como un nuevo seguidor, comentario o like. Por otro lado, no importante que la aplicación sea responsive, podemos suponer que inicialmente solo se va a usar en celulares y nada de tablets.”

## **Entregas**

- Primera Entrega
  - Pantallas (60%), navegación, MVVM, entidades, diagramas de clase, diagrama de entidad – relación, documento de requisitos, mockups de toda la aplicación, repositorio en Github, Información general de la aplicación (nombre, icono, paleta de colores, tipografía)
- Segunda Entrega
  - Pantallas (30%), creación de API REST y BD SQL, consumo de API REST con retrofit, generación de BD NSQL en Firebase, Consumo de datos en Firestore, autenticación en Firebase, uso de Intents, manejo de permisos, manejo y consumo de recursos (imágenes) en firestore, uso de patrones de software (capas, repositorio, MVVM, fachada, inyección de dependencias, etc).
- Tercera Entrega
  - Pantallas (10%), pruebas automatizadas de la aplicación, uso de otro framework(Swift o Flutter) para creación de al menos 2 pantallas adicionales y consumo de datos, manejo de notificaciones a través de firebase.
  - Opcionalmente dependiendo del desarrollo del curso a lo largo del semestre se podrán ver temas adicionales como IA en dispositivos móviles, accesibilidad, responsive, Kotlin multiplataforma, manejo de mapas, entre otros.
  - Video de 5 minutos promocional de la aplicación
  - Presentación promocional de máximo 10 minutos de la aplicación.

## **Metodología**

Se manejará una metodología ágil basada en Scrum. Esto quiere decir que todas las semanas habrá Sprints en los cuales es obligatorio que los grupos muestren adelantos. La meta de cada Sprint se definirá en clase, y se busca que al momento de la entrega el usuario no tenga trabajo acumulado.

## **Calificación**

La nota de los grupos se dividirá en 2 partes (2.5 unidades C/U). La primera será la calificación del cliente, que se calculará a partir de la entrega de cada Sprint. La segunda será la calificación del product manager el cuál revisa que todos los requerimientos a nivel funcional de la aplicación se cumplan.

### **Calificación del cliente:**

Cada semana se realizarán entre una y dos entregas en formato de sprint donde los grupos de trabajo deben mostrar adelantos FUNCIONALES de su trabajo.

El cliente se fija de manera general en temas funcionales, UX, UI y CX, para dar su calificación por sprint.

Al final de semestre la calificación del cliente aporta para distinguir cuál ha sido el mejor proyecto durante todo el curso.

### **Calificación del Product owner:**

Este revisará al final de cada corte a nivel funcional y de código como va el producto general a entregar. Se usará un enfoque de calificación “substractiva” donde cada uno de los grupos empieza con las 2.5 unidades y por cada error cometido se van subtrayendo decimas de la nota final.

Algunas de las penalizaciones son:

- Código no comentado
- Malos nombres de variables
- No hay validación de los formularios
- Problemas de rendimiento de la aplicación
- No se cumplen con los requisitos funcionales propuestos
- Código duplicado (puede ser por mal uso de funciones o de componentes)
- No usar los patrones de diseño mostrados en clase
- Mal diseño (de base de datos, diagrama de clases, navegación)
- Usar paquetes para organizar el proyecto de manera coherente
- Entre otros...