

Proyecto 2

Introducción

El objetivo de este proyecto es familiarizarse con el desarrollo de aplicaciones móviles, mediante la creación de una aplicación móvil. La idea para la aplicación deberá ser propuesta por cada equipo, y el alcance será validado con el profesor.

Requerimientos

1. Desarrollo en Android nativo
2. Deben usar Material Design para el app de Android
3. Hacer un landing page de mercadeo para el app y publicarlo en Internet
4. Desarrollo de backend propio (no BaaS), que sea accesible por dominio público. Pueden usar cualquier framework para el Backend, excepto Python/Django.
5. El backend debe tener un backoffice web de administración, con su propia autenticación
6. Deben usar user login, tanto en backoffice como en aplicación
7. Deben manejar autenticación y autorización a nivel de API
8. Conexión con al menos dos APIs externos (sin contar el backend propio)
9. Usar una de las siguientes características del teléfono: GPS, acelerómetro, cámara, compás
10. El app debe tener al menos 12 activities
11. Incluir al menos:
 - a. 2 patrones de despliegue de datos
 - b. 1 patrón de búsqueda
 - c. 1 patrón de filtro
 - d. 1 patrón de ordenamiento
 - e. 1 patrón de navegación
12. Deben manejar métricas de uso con Mixpanel y Fabric
13. Uso de AWS (EC2, S3 y Route53) para infraestructura (10% adicional en la nota de la progra)

Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

La documentación externa deberá incluir:

- Propósito: descripción del app, requerimientos
- Funcionalidades

- UX, UI (wireframes)
- Descripción de diseño de alto nivel
- Descripción de los web services (descripción, url, params, método http)
- Interacción con sistemas externos

Aspectos administrativos

- La tarea vale un 40% de la nota del curso
- La tarea se hará en grupos de 2-3 personas.
- Los grupos deberán subir el código y la documentación de sus respectivas tareas a un repositorio en Github, de manera que el profesor pueda ver las contribuciones que las diferentes personas hacen al proyecto. La idea es que apenas empiecen a desarrollar la tarea, suban las contribuciones al repositorio, y no esperar a tener todo el código listo para subirlo.
- Deberán enviar un correo a las direcciones **andreifu.tec@gmail.com**, en donde indiquen el url del repositorio de Github en donde se encuentra el código y la documentación de la tarea. El asunto del correo enviado tendrá el siguiente formato: **IC-8041 TP2-nombrea1-nombre2-nombre3**. Las tareas que no sean entregadas por medio de Github tendrán nota cero.
- Las tareas deberán ser revisadas con el profesor. Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente. La nota de la revisión es individual, el resto de la nota es grupal.
- El código entregado debe ser 100% original. En caso de probarse algún tipo de fraude en la elaboración de la tarea, se aplicarán todas las medidas correspondientes, según el reglamento del TEC, incluyendo una carta al expediente.
- Fecha de entrega:
 - Definición de idea a trabajar: jueves 10 de mayo, 7:30 a.m. Deben enviar un correo al profesor con el asunto **IC-8041 TP2-Propuesta nombre1-nombre2-nombre3**, y deben incluir únicamente un link a un documento de Google Slides que incluya los wireframes de la aplicación, para definir el alcance.
 - Entrega final: martes 5 de junio, 7:30 a.m.

Evaluación

Criterio	Peso	Excelente (100%)	Adecuado (70%)	Regular (30%)	Deficiente (0%)
Documentación interna	3%	Los comentarios de las clases y funciones explican claramente lo que hacen. Los nombres de funciones, variables y clases son claros, y permiten entender perfectamente el código.	Una o dos secciones podrían tener mejores comentarios, tanto en clases como en funciones.	Hay muchas secciones del código sin comentar. correctamente Muchos de los comentarios no son lo suficientemente claros. Usan nombres de variables poco claros.	El código fuente no está comentado del todo.
Documentación externa	7%	Usan buena redacción. La documentación tiene todas las secciones. Las secciones tienen suficiente detalle para entender.	Errores de redacción menores. La documentación tiene todas las secciones, pero falta detalle en al menos una de las secciones.	Errores de redacción frecuentes. La documentación no tiene todas las secciones, o falta profundizar en alguna de las secciones.	No entregan documentación.
Funcionalidad backend	35%	Todas las funcionalidades del backend están implementadas correctamente	Faltan de 1 a 2 funcionalidades menores de implementar, o falta 1 funcionalidad mayor de implementar	Faltan de 3 a 4 funcionalidades menores de implementar, o faltan 2 funcionalidades mayores de implementar	Faltan 5 o más funcionalidades menores de implementar, o 3 o más funcionalidades mayores
Funcionalidad aplicación	45%	Todas las funcionalidades de la aplicación están implementadas correctamente	Faltan de 1 a 2 funcionalidades menores de implementar, o falta 1 funcionalidad mayor de implementar	Faltan de 3 a 4 funcionalidades menores de implementar, o faltan 2 funcionalidades mayores de implementar	Faltan 5 o más funcionalidades menores de implementar, o 3 o más funcionalidades mayores
Revisión	10%	Responden adecuadamente a cada una de las preguntas hechas durante la revisión. Demuestran conocimiento avanzado de la materia	Muestran conocimiento medio de la materia y de la solución entregada.	Muestran conocimiento muy básico de la materia. No responden correctamente algunas de las preguntas.	No asisten a la revisión.