

**Curso:** Desarrollo de aplicaciones móviles

**Sesión:** Laboratorio

**Número:** 4

**Ciclo:** 2020-2

**Fecha:** Martes 17 de Noviembre

**Consideraciones generales:**

- Equipos de trabajo: 3 alumnos
- Tiempo: 2 horas
- GIT:
  - Debe crear un repositorio en github donde integre todo el funcionamiento
  - Indicar su repositorio en PAIDEIA
  - En el repositorio git deben existir commits de todos los integrantes del grupo.
  - Si un integrante del grupo no tiene commits en el repositorio git o sus datos no son suyos, no tendrá nota.
  - Se revisará la rama master
- Nombre del proyecto: **Lab4GX**
  - Donde X es su número de grupo

Ejercicio:

## Caso: App de Democracia

Debido a los recientes hechos políticos en el país de Departamentos Unidos de Sudamerica ( DDUU ) , se ha decidido contribuir con la preservación de la democracia, para lo cual el gobierno de dicho país ha seleccionado a los grupos del curso de TEL306 de la PUCP para que realicen una aplicación que consulte a los ciudadanos respecto a diversas preguntas y las decisiones sean tomadas de acuerdo al consenso popular.

Dicho país le ha habilitado una API REST en sus propios datacenters para evitar que los hackers puedan extraer o modificar la información de los ciudadanos.

La lógica de esta primera versión de la aplicación Android es simple:

- Todos los usuarios pueden registrarse, pero solamente los usuarios debidamente logeados pueden ver las preguntas.
- Los datos de los usuarios logeados deben guardarse en la memoria del telefono en un archivo json que contenga esta información. Su aplicación debe requerir los permisos necesarios para realizar esto.
- Se debe mostrar una lista con todas las preguntas disponibles
- El gobierno a través de su API proporcionará las preguntas y los usuarios podrán responder a ellas.
- Toda pregunta tendrá varias opciones
- Los usuarios que deseen responder alguna pregunta solamente pueden seleccionar una de las opciones.
- Cuando se responde una pregunta, se muestra inmediatamente una vista con las estadísticas de dicha pregunta (no se debe mostrar las estadísticas antes de responder una pregunta).
- Una vez que un usuario haya respondido una pregunta no puede cambiar de opinión, por lo que cuando el usuario consulte una pregunta que ya ha respondido no se le deben mostrar las opciones sino un gráfico de las estadísticas de dicha pregunta (opcion A: X%, opcion B: Y%, opcion C: Z%).

Dicho gobierno puntuará las aplicaciones que los grupos realicen de acuerdo a los siguientes criterios:

### 1. Usa buenas prácticas de codificación (3 ptos)

- Se tiene un ícono en la app.
- El paquete base debe ser **pe.pucp.dduu.tel306**
- Todos los strings deben estar en xml.
- Usar fragments para las diversas activities que considere necesarias.
- Usar la API REST provista
- Buen diseño de UX para la app
- Se solicitan los permisos necesarios para el funcionamiento de la aplicación

## 2. Usa fragments en el inicio de sesión / registro de usuario (8 ptos)

- Fragment de inicio de sesión (4 ptos)

Este fragment se muestra **solamente si la aplicación no presenta los datos de usuario en su memoria local** (un archivo json).

El fragment de inicio de sesión debe contar con 2 campos para el mail y la contraseña, además de tener un botón para iniciar sesión que debe llamar al endpoint de la API encargado de realizar el login (**POST /api/v1/users/login**) con el email y la contraseña. Y si esta API retorna un resultado favorable (los datos de usuario), se procede a la pantalla de todas las preguntas y se guardan los datos de usuario en el archivo json.

También debe tener un botón de registrar usuario que cambie este fragment por el fragment de registro de usuario.

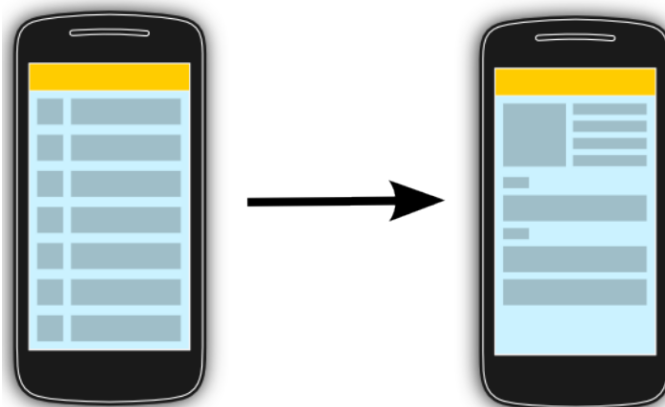
- Fragment de registro de usuario (4 ptos)

Este fragment se muestra **solamente si la aplicación no presenta los datos de usuario en su memoria local** (un archivo json).

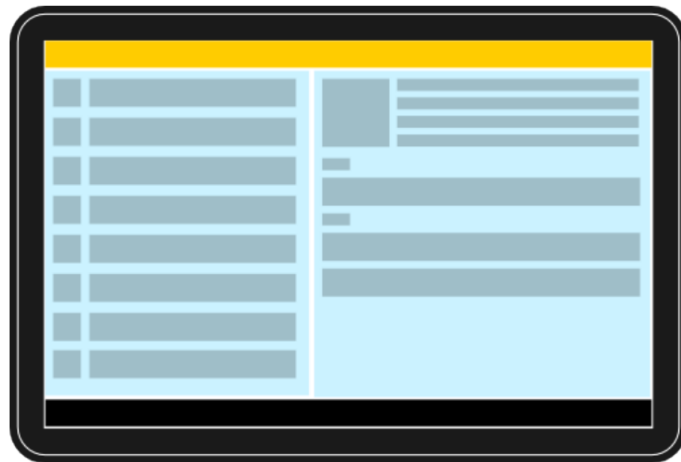
El fragment de registro de usuario debe contar con 3 campos (nombre del usuario, mail y contraseña) y ser enviado al endpoint de registro de usuarios (**POST /api/v1/users/new**), si es que esta API retorna un resultado favorable (los datos de usuario), se procede a la pantalla de todas las preguntas y se guardan los datos de usuario en el archivo json.

También debe tener un botón de iniciar sesión que cambie este fragment por el fragment de inicio de sesión.

## 3. Usa fragments en las preguntas y en el detalle de las preguntas (9 ptos)



Si hubiera más espacio disponible (como en una tablet)



- Fragment de preguntas (listado de todas las preguntas) **(4 ptos)**

Valide que el usuario ha iniciado sesión (mediante el archivo json).

Se debe mostrar una lista de todas las preguntas que nos manda el endpoint de la API (**GET /api/v1/questions**). Al hacer click a una de ellas se debe mostrar el fragment de detalle de pregunta.

- Fragment de detalle de una pregunta **(5 ptos)**

Valide que el usuario ha iniciado sesión (mediante el archivo json).

Verificar si el usuario ya ha respondido a una pregunta (**GET /api/v1/answers/detail?questionid={questionid}&userid={userid}**)

Si el usuario aún no ha respondido esa pregunta, presentarle las opciones obtenidas en (**GET /api/v1/questions/{idquestion}**) y su respuesta enviarla como (**POST /api/v1/questions/{idquestion}/answer**) enviando un json con el id de usuario y id de respuesta.

En caso ya haya respondido, se debería mostrar las estadísticas de las respuestas (**GET /api/v1/answers/stats?questionid={questionid}**)

En la versión base de la app (si no desea realizar las preguntas de bonus) bastará con mostrar los resultados de las respuestas en este fragment pero sin necesidad de hacer gráficos o estadísticas.

Ejm:

Opcion 1 -> 20%

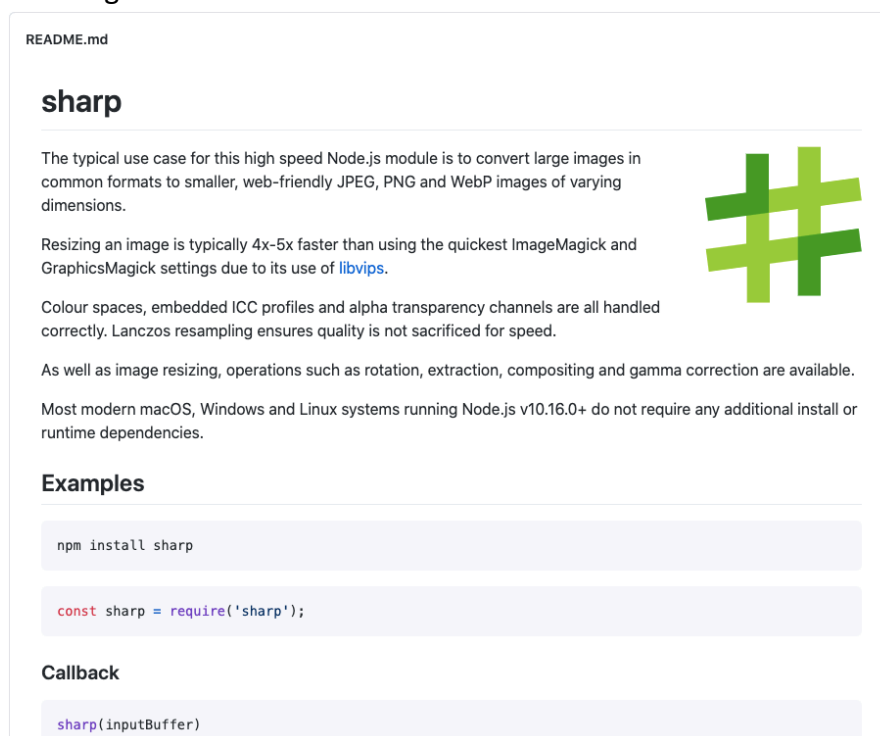
Opcion 2 -> 30%

...

Además, se considerarán los siguientes puntos bonus opcionales :

4. El Repositorio Git debe tener un buen README.md que explique claramente el propósito de la aplicación, algunos screenshots de su app, propuestas de mejora de la app, del API, etc. (0.5 ptos en español / 1 pto en inglés)

Puede usar como ejemplo los README.md de los repositorios populares en github. Imagen referencial.



5. Validación de datos para fragments de inicio de sesión / registro de usuario (0.5 ptos)

Se puede utilizar expresiones regulares (regex) internamente para validar diversos tipos de entrada de texto. En este caso, que el nombre de los usuarios consista de 2 palabras, sin números ni caracteres especiales (Ejm: Juan Perez).

Sugerencia: utilice regexr.com para validar su cadena regex e investigue como implementar regex en android + java.

6. Botón de cerrar sesión que borre el archivo json en el que se guardan los datos de usuario (0.5 ptos)

Puede colocarlo en cualquier parte visible de su aplicación, pero su función es borrar el archivo json (o su contenido) e invocar al primer activity con los fragments de inicio / registro.

#### 7. Fragment de estadísticas de respuestas de una pregunta (1 pto)

En este caso, cuando el usuario ya haya respondido a una pregunta (y puesto que ya no puede cambiar su respuesta), se puede mostrar un gráfico de estadísticas para visualizar los datos.

Sugerencia: use la librería AnyChart, pero investigue la forma de usarla dentro de un fragment.

<https://github.com/AnyChart/AnyChart-Android>

<https://github.com/AnyChart/AnyChart-Android/wiki/Getting-started>

