

SISTEMAS OPERATIVOS AVANZADOS	Curso: 1123 y 625
<b>Trabajo Práctico Android</b>	
AUTOR: Docentes Sistemas Operativos Avanzado	EDICIÓN: BUENOS AIRES, 2020

## Enunciado del Trabajo Práctico de Android

### Modalidad de trabajo:

- El trabajo podrá ser realizado en forma individual o en grupo de 2 integrantes.

El trabajo práctico consistirá en proponer una idea de aplicación que involucre los diferentes conceptos vistos en clase. Dicha aplicación deberá cumplir con los siguientes requisitos:

- 1) Generar un repositorio público en Github para la entrega del trabajo práctico. El repositorio deberá estar compuesto por los siguientes directorios:
  - a. **CODIGO**
  - b. **EJECUTABLE**

En el directorio CODIGO se deberá colocar el código fuente de la aplicación. Luego, en el directorio EJECUTABLE se deberá guardar el **apk** de la aplicación (archivo binario del programa)

- 2) La aplicación Android a desarrollar deberá cumplir con los siguientes requisitos:
  - a. Se deberá implementar un sistema con Registro de usuario y Login haciendo peticiones a un servidor, a través de la API desarrollada por la cátedra.
  - b. Se deberá validar el estado de la conexión de internet, en las pantallas de Registro y Login. En caso de que no haya conexión, se deberá mostrar mensaje de error al usuario.
  - c. Se deberá implementar alguna de las ejecuciones en Background vistas en clase.
  - d. Se deberán hacer uso de al menos dos sensores propios del dispositivo móvil.
  - e. La aplicación deberá registrar en el servidor diferentes eventos ocurridos durante su ejecución, como por ejemplo: Actividad de sensores, Login, Ejecuciones en Background, Broadcast, etc. Para ello se deberá utilizar la API desarrollada.
  - f. Al escuchar los distintos eventos de un determinado sensor, o conjunto de eventos, los datos capturados deberán ser mostrados en un listado por pantalla. Esta información deberá ser guardada en forma persistente, usando **SharedPreferences**. Posteriormente los datos mostrados deberán volverse a cargar en el listado, en cada inicio de sesión.
  - g. La aplicación deberá tolerar diferentes fallos:
    - i. Pruebas de respuesta fluida.
    - ii. Liberación de recursos.
    - iii. Cambios de estado.
    - iv. Errores de conexión.
  - h. Opcionalmente podrán implementar servicios externos como Firebase o desarrollar uno propio.

### Aclaraciones:

- **El alumno deberá informar previamente la idea de negocio a realizar, con el objetivo de validar que cumpla con los requisitos previos al desarrollo.**
- La aplicación debe funcionar correctamente. Además, se va a poder comprobar que no este Harcodeado el post, chequeando el código fuente del tp de los alumnos.
- **Se dará un plazo hasta el 13 de mayo para entregar las propuestas del programa (negocio) que va a desarrollar cada grupo.**
- **Para la entrega final tendrá tiempo de entregar el trabajo práctico hasta la 1ra semana de junio.**
- Versiones de Android Studio recomendadas con las que se puede trabajar son 2.4 o 3.5.2

SISTEMAS OPERATIVOS AVANZADOS	Curso: 1123 y 625
<b>Trabajo Práctico Android</b>	
AUTOR: Docentes Sistemas Operativos Avanzado	EDICIÓN: BUENOS AIRES, 2020

## Estructura de peticiones API REST

En el trabajo práctico se deberán enviar mensajes al servidor usando peticiones API que cumplen el protocolo REST. En este sentido existen tres tipos de peticiones de mensajes POST, que se deberán ser utilizadas para poder realizar lo siguiente.

- a. Registrar un usuario/alumno en la base de datos del Web services
- b. Verificar los datos de los alumnos, que se encuentran registrados en la base de datos del web services, para poder hacer un Login al sistema
- c. Registrar la ocurrencia de distintos eventos en el servidor.

La utilización de estos mensajes ya fue explicado anteriormente. No obstante a continuación se describen el formato, y los datos que se deben enviar y recibir en cada una de las distintas peticiones REST al servidor.

### 1) REGISTRO DE USUARIO/ALUMNO

- URI: <http://so-unlam.net.ar/api/api/register>
- Tipo de Petición: **POST**
- Estructura de datos (content-type): **application/json**
- Datos del Request:

**Body:**

```
{
  "name": String(255),
  "lastname": String(255),
  "dni": Int(20),
  "email": String(255) Email,
  "password": String(255), MIN(8), //Debe tener al menos 8 caracteres
  "commission": Int(20),
  "group": Int(20) //el número se la asignará los docentes
}
```

- Datos del Response:

**Success:**

```
{
  "state": "success",
  "token": "$2y$10$FhkKBS/E37cMyf6N91TCkuaJJoJH4Z1hy5o.fb8ZSNzuBvEsRSIKC"
}
```

**Error:**

```
{
  "state": "error",
  "msg": "Error en los parámetros enviados"
}
```

### 2) LOGIN DE USUARIO/ALUMNO

- URI: <http://so-unlam.net.ar/api/api/login>
- Tipo de Petición: **POST**
- Estructura de datos (content-type): **application/json**
- Datos del Request:

**Body**

```
{
  "name": String(255),
  "lastname": String(255),
  "dni": Int(20),
  "email": String(255) Email, //se valida solamente el mail y el password
  "password": String(255), MIN(8), //se valida solamente el mail y el password
}
```

## Trabajo Práctico Android

```

        "commission": Int(20),
        "group": Int(20) //el número se la asignará los docentes
    }
}

```

- Datos del Response:  
**Success:**

```

{
    "state": "success",
    "token": "$2y$10$FhkKBS/E37cMyf6N9lTCkuaJJoJH4Z1hy5o.fb8ZSNzuBvEsRSIKC"
}

```

**Error:**

```

{
    "state": "error",
    "msg": "Error de autenticación"
}

```

3) REGISTRAR EVENTO

- URI: <http://so-unlam.net.ar/api/api/event>
- Tipo de Petición: **POST**
- Estructura de datos (content-type): **application/json**
- Datos del Request:

**Header**

```

"token": "$2y$10$FhkKBS/E37cMyf6N9lTCkuaJJoJH4Z1hy5o.fb8ZSNzuBvEsRSIKC"

```

**Body**

```

{
    "type_events": String(255), //Debe indicarse si es un evento de un sensor, Login, Service,
                                //Broadcast, etc
    "state": "ACTIVO/INACTIVO",
    "description": String(255) //Poner un descripción de lo sucedido en el evento registrado
}

```

- Datos del Response:

```

Success:
{
    "state": "success",
    "event": {
        "type_events": "Broadcast Receiver",
        "state": "Activo",
        "description": "Broadcast que detecta la desconexión 3G",
        "group": 123
    }
}

```

**Error:**

```

{
    "state": "error",
    "msg": "Error de autenticación"
}

```

```

{
    "state": "error",
    "msg": "Error en los parámetros enviados"
}

```