



PERÍODO 2021



## TRABAJO AUTÓNOMO 5

**DESARROLLO DE UNA APLICACIÓN MÓVIL  
USANDO COMPONENTES AVANZADOS Y  
REPOSITORIOS DE GITHUB**

GRUPO#2

DEMERA CARRANZA, MEJIA BEDOR, RENDON PANTOJA, RONQUILLO MANOSALVAS

ELABORADO POR: ADRIANA COLLAGUAZO JARAMILLO  
MATERIA: PROGRAMACIÓN DE SISTEMAS TELEMÁTICOS  
CARRERA DE INGENIERÍA EN MECATRÓNICA  
FIEC-ESPOL

<b>Trabajo Autónomo:</b>	<b>Unidad 3 Control remoto de sistemas telemáticos</b>
<b>Objetivo de Aprendizaje:</b>	Desarrollar interfaces de usuario en entornos de desarrollo web y móvil para el control remoto de los sistemas telemáticos.
<b>Recursos:</b>	Android Studio, GIT (software). Github (online).
<b>Duración:</b>	8 horas

## INSTRUCCIONES

- El formato del trabajo tiene habilitado recuadros para que llenen las respuestas.
- Los trabajos se reciben hasta la fecha planificada en el SIDWeb.
- Coloque el nombre del archivo así "PST\_TAA\_GrupoB\_Apellido1\_ApellidoN", siendo A el número del trabajo, B el número del grupo, N el último apellido del integrante del grupo.
- Una vez que haya desarrollado el trabajo, cada integrante del grupo contestará la encuesta de evaluación de los trabajos autónomos ingresando al enlace. → <https://forms.gle/d2p61cJrjd2ZFqsu8>

## INTRODUCCIÓN

Android Studio es un IDE, es decir, un entorno de desarrollo integrado oficial para el desarrollo apps para el SO Android basado en IntelliJ IDEA. Entre sus funciones destacan la compilación flexible basada en Gradle; emulación rápida y cargado de funciones; proporciona un entorno unificado de desarrollo para cualquier versión de Android; es compatible con C++, NDK, Google Cloud Platform y permite la generación de APK.<sup>1</sup>

Este IDE puede además integrarse con GitHub, el cual es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. Esta plataforma es usada por desarrolladores a nivel global para facilitar el trabajo en equipo a través de sus opciones de clonación, merge, etc. Y al mismo tiempo compartir de manera libre su trabajo contribuyendo a la mejora continua de software.<sup>2</sup>

Desarrolle un aplicativo móvil usando componentes avanzados como menú, y cargue el código fuente en un repositorio de GitHub.

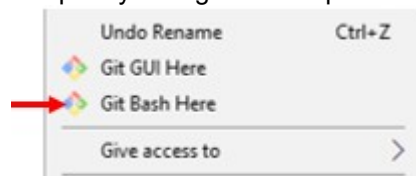
## ACTIVIDADES

### Paso 0: Creación de cuenta en Github.

1. Crearse una cuenta en <https://github.com>.
2. Instalar Git Bash en el enlace <https://git-scm.com/downloads>. En caso de que no se disponga de GIT CLI (git bash), también puede utilizar CMD de Windows/Ubuntu.

### Paso 1: Crear un repositorio.

1. Creamos un proyecto de Android Studio, el cual vamos a alojar en nuestro repositorio de github.
2. Dentro de la carpeta del proyecto, abra la línea de comandos de GIT (GIT CLI). Podemos encontrarlo, dando clic derecho dentro de la carpeta y escogemos la opción "GIT BASH HERE".



<sup>1</sup> Android Studio <https://developer.android.com/studio/intro?hl=es> <sup>2</sup> GitHub <https://github.com>

- Para probar que GIT ha sido instalado correctamente, utilice el comando "git --version".

```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\acollag>git --version
git version 2.13.3.windows.1
```

GIT en línea de comandos de Windows

```
acollag@GGIECDCWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (master)
$ git --version
git version 2.13.3.windows.1
```

GIT BASH propia

- Para crear un nuevo repositorio, utilice el siguiente comando "git init".

```
acollag@GGIECDCWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (master)
$ git init
Initialized empty Git repository in C:/Users/acollag/AndroidStudioProjects/PST_TA5_G#/.git/
```

Esto creará un archivo oculto [.git] para el manejo del repositorio y nos ubicará directamente en la rama "master"

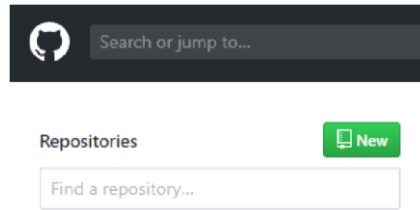
- Agregamos todos los archivos del proyecto a nuestro **repositorio local** con el comando: "git add --all".

```
acollag@GGIECDCWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (master)
$ git add --all
warning: LF will be replaced by CRLF in gradle/wrapper/gradle-wrapper.properties.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in gradlew.
The file will have its original line endings in your working directory.
```

- Ahora realizamos un **commit**, esto realizará nuestros cambios permanentes en el repositorio local. Pero debemos asignarle un mensaje [-m "mensaje"] para indicar los cambios que hemos realizado.

```
acollag@GGIECDCWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (master)
$ git commit -m "Commit inicial"
[master (root-commit) 57bedc8] Commit inicial
29 files changed, 478 insertions(+)
create mode 100644 .gitignore
create mode 100644 app/.gitignore
create mode 100644 app/build.gradle
create mode 100644 app/proguard-rules.pro
create mode 100644 app/src/androidTest/java/com/example/acollag/pst_ta5_g/ExampleInstrumentedTest.java
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/example/acollag/pst_ta5_g/MainActivity.java
create mode 100644 app/src/main/res/layout/activity_main.xml
create mode 100644 app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-hdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-mdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-xhdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 app/src/main/res/mipmap-xxhdpi/ic_launcher_round.png
create mode 100644 app/src/main/res/values/colors.xml
create mode 100644 app/src/main/res/values/strings.xml
create mode 100644 app/src/main/res/values/styles.xml
create mode 100644 app/src/test/java/com/example/acollag/pst_ta5_g/ExampleUnitTest.java
create mode 100644 build.gradle
create mode 100644 gradle.properties
create mode 100644 gradle/wrapper/gradle-wrapper.jar
create mode 100644 gradle/wrapper/gradle-wrapper.properties
create mode 100644 gradlew
create mode 100644 gradlew.bat
create mode 100644 settings.gradle
```

- Creamos un repositorio en línea. Ahora usaremos Github (Requerira una cuenta gratuita). Del lado superior izquierdo, encontrara el botón "NEW".



7. La información requerida para crear un repositorio, se muestra a continuación:

<b>Nombre del repositorio</b>	El nombre de nuestro repositorio como será publicado en línea
<b>Descripción (Opcional)</b>	Descripción sobre lo que realiza nuestro proyecto
<b>Tipo</b>	Público o privado (para saber si es visible en línea)
<b>Archivo Readme</b>	Archivo inicial del repositorio. Agregamos indicaciones a otros programadores
<b>Agregar. gitignore</b>	Archivo para seleccionar los archivos que no queremos subir a nuestro repositorio
<b>Licencia</b>	Tipo de licencia: OpenSouce, MIT, Apache, etc

Debe asignarle el siguiente nombre a su repositorio "PST\_TA5\_G#", reemplazando el "#" por su número de grupo, por ejemplo: "PST\_TA5\_G1".

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner: acollaguazo / Repository name \*: PST\_TA5\_G# ✓

Great repository names are Your new repository will be created as PST\_TA5\_G- but probable-succotash?

Description (optional): Este repositorio es creado para el desarrollo del proyecto de programación de sistemas telemáticos.

☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ


Create repository

8. Una vez ingresados todos los campos, damos clic en "Create repository".



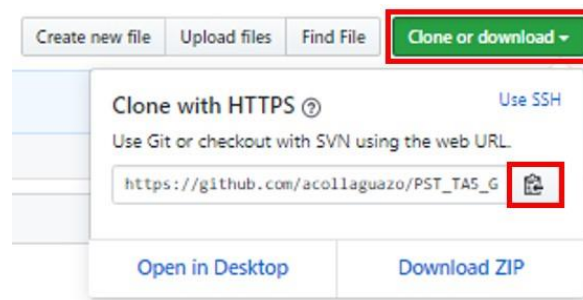
Vista de mi repositorio vacío.

9. Damos clic en el botón verde “clone or download” donde estará visible el **URL** para su manejo y

presionamos en para  copiarlo.

Para obtener el repositorio en línea, obtenemos la rama de externa con el comando:  
\$ git remote add origin [web URL del repositorio]

```
acollag@GGIECDOCWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (master)
$ git remote add origin https://github.com/acollaguazo/PST_TA5_G-.git
```



10. Ahora tendremos que obtener la rama y publicar los cambios:

\$ git pull origin master [obtiene la rama externa de Github (master)]

\$ git push origin master -f [publica el proyecto local (-f para forzar los cambios)]

11. Repetir los pasos 4, 5 para actualizar los cambios realizados y el comando git push origin master.

### PREGUNTAS DE INVESTIGACIÓN:

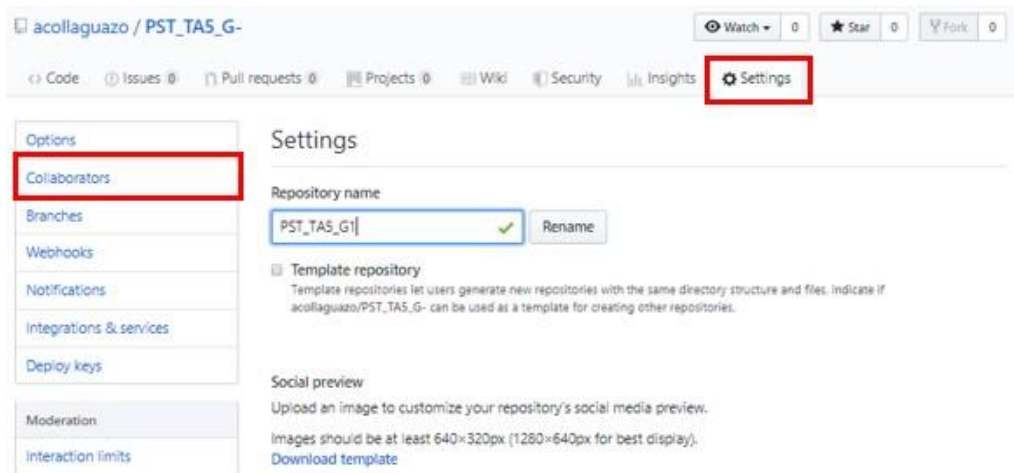
- ¿Qué otro tipo de servicios en línea (como GITHUB) existen?
- ¿Para qué sirve el archivo .gitignore y como se utiliza?
- ¿Qué limitaciones tiene GITHUB?



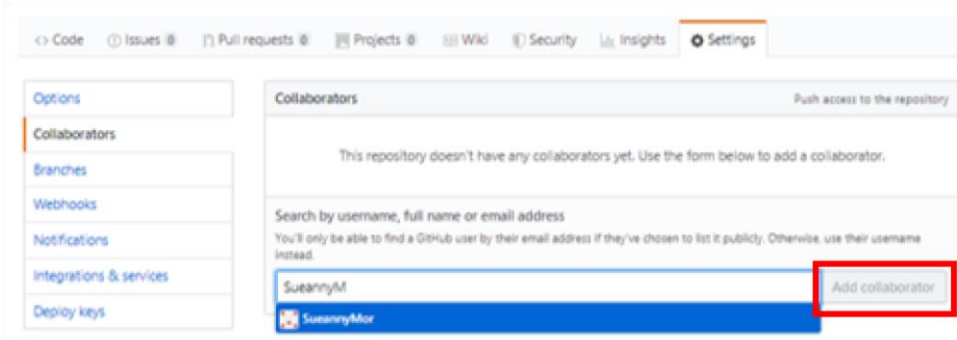
4. ¿Qué es una rama?
5. ¿Con cuál nombre, comúnmente, se le denomina a la rama principal?
6. ¿Cuál es el link de su repositorio?
7. ¿Cómo utilizo un repositorio público (utilizando el comando git clone)?

## Paso 2: Invitar a otros miembros del grupo a mi proyecto (incluya a todos los miembros del grupo)

1. Para habilitar la modificación a otros miembros de mi grupo, debemos darle acceso. [Incluso si el proyecto es libre, solo pueden modificarlo quienes han sido invitados]. Para esto de clic en el tab "Settings/Configuración" > Collaborators / Colaboradores



2. Buscamos y agregamos a otros usuarios (usuario de github).



3. Una vez agregados, es necesario aceptar las invitaciones para poder realizar PUSH (cambios al repositorio). Puede revisar las invitaciones en la campana a lado del usuario.



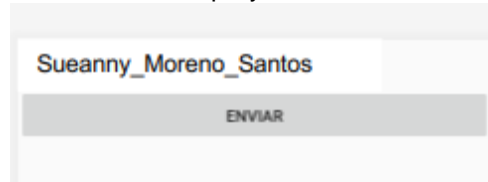
## Paso 3: Crear una rama [branch] (Trabajo individual)

1. Tenemos el proyecto principal en master, cualquier otro cambio puede ser realizado sin dañar el proyecto principal. Utilizamos el comando:  
git checkout -b "nombre\_rama"

Para este taller, crearemos una rama de la siguiente forma: "nombre\_apellido1\_apellido2"

```
aco1lag@GGIECD0CWRK054 MINGW64 ~/AndroidStudioProjects/PST_TA5_G# (SueannyMor)
$ git checkout -b "Sueanny_Moreno_Santos"
Switched to a new branch 'Sueanny_Moreno_Santos'
```

2. Ahora realizamos algunos cambios en nuestro proyecto local.



3. Estos cambios son únicamente reflejados dentro de nuestra rama. Ahora subimos los cambios, para ello realizamos los siguientes comandos.

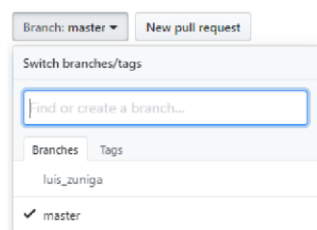
Git add --all	Agrega todos los cambios a nuestra rama.
Git commit -m "cambios a mi rama"	Agrega un commit en mi rama, indicando los cambios que realice.
Git push origin [nombre rama]	Subimos los cambios al repositorio (pero solo dentro de la página).

```
zuniga@DESKTOP-JDQFA20 MINGW64 ~/Desktop/AMST_3 (luis_zuniga)
$ git add --all

zuniga@DESKTOP-JDQFA20 MINGW64 ~/Desktop/AMST_3 (luis_zuniga)
$ git commit -m "cambios a mi rama"
[luis_zuniga 5cb9235] cambios a mi rama
8 files changed, 202 insertions(+), 101 deletions(-)
create mode 100644 .idea/vcs.xml
rewrite app/src/main/res/drawable/ic_launcher_background.xml (97%)

zuniga@DESKTOP-JDQFA20 MINGW64 ~/Desktop/AMST_3 (luis_zuniga)
$ git push origin luis_zuniga
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 2 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (18/18), 2.00 KiB | 78.00 KiB/s, done.
Total 18 (delta 9), reused 0 (delta 0)
```

4. Podemos revisar todas las ramas dentro de Github (así mismo podemos cambiar entre ramas para revisar diferentes versiones de código).

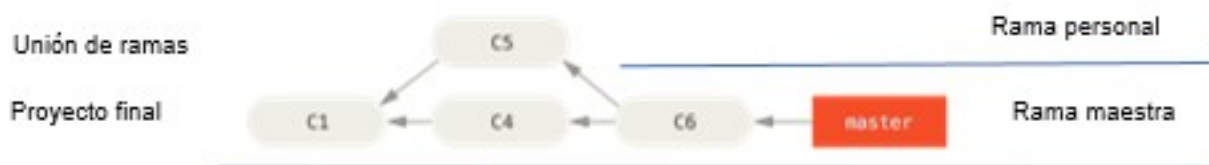


## PREGUNTAS DE INVESTIGACIÓN

1. ¿Qué utilidades tiene el manejo de ramas?
2. ¿Qué tipos de conflictos puede ocurrir durante el manejo y creación de ramas?
3. ¿Cuál es la diferencia entre **checkout** y **checkout -b**?
4. ¿Qué es la herramienta **lint**?
5. Escriba al menos **5 buenas prácticas de estándares de programación** para creación de aplicaciones móviles en Android Studio/Java.
6. Investigue acerca de los tipos de aplicaciones móviles, su principal característica, las tecnologías con las que se programan y ejemplos de aplicaciones de cada tipo.

### Paso 4: Unir ramas al proyecto principal [branch].

Las ramas funcionan como proyectos paralelos del proyecto principal, pero para avanzar con el proyecto es necesario unir las ramas una vez han sido probadas.



1. En caso de realizar algún cambio en el repositorio maestro.
  - a. Git fetch origin master (obtiene todos los cambios realizados en master)
2. Nos cambiamos a la rama principal.
  - a. Git checkout master
3. Traemos los cambios realizados en la rama única.
  - a. Git merge [nombre rama]

```

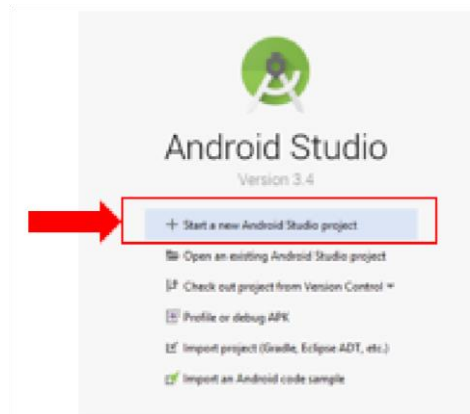
zuniga@DESKTOP-JDQFA20 MINGW64 ~/Desktop/AMST_3 (master)
$ git merge luis_zuniga
Updating 9430617..5cb9235
Fast-forward
 .idea/vcs.xml | 6 +
 app/src/main/AndroidManifest.xml | 6 +-
 .../res/drawable-v24/ic_launcher_foreground.xml | 12 +-
 .../main/res/drawable/ic_launcher_background.xml | 236 ++++++
 app/src/main/res/layout/activity_main.xml | 23 +-
 app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml | 4 +-
 .../res/mipmap-anydpi-v26/ic_launcher_round.xml | 4 +-
 gradle.properties | 4 -
 8 files changed, 198 insertions(+), 97 deletions(-)
 create mode 100644 .idea/vcs.xml
    
```

*Nota: Esto indica los archivos que han sido modificados.*

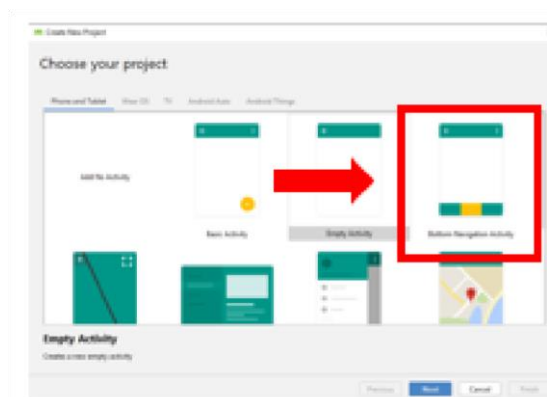
### Paso 5: Crear un nuevo proyecto en Android Studio

1. Al abrir Android Studio, podemos crear, abrir o importar proyectos. Seleccione "Start a new Android Studio project".

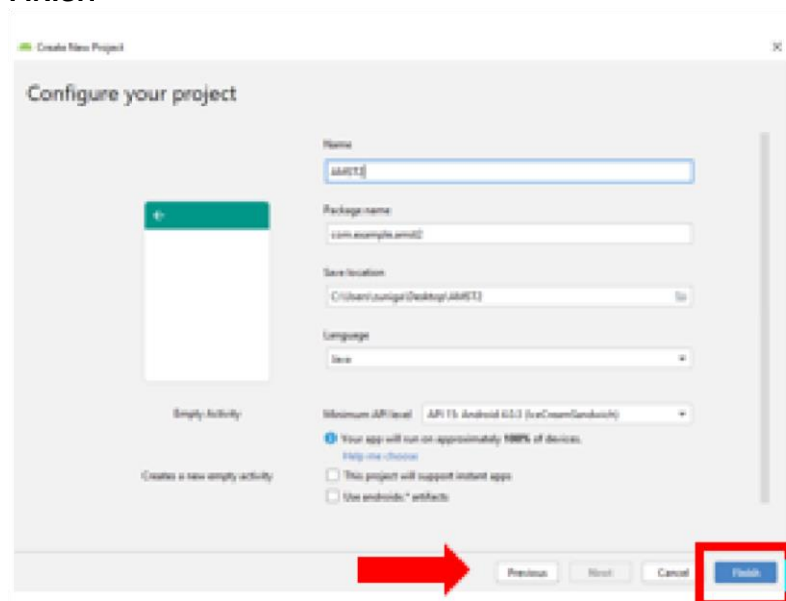




2. Seleccionar el tipo de proyecto: Para esta práctica escogeremos la pestaña **Phone and Tablet > Empty Activity**. Otro tipo de actividades viene por defecto con componentes no necesarios para este taller.



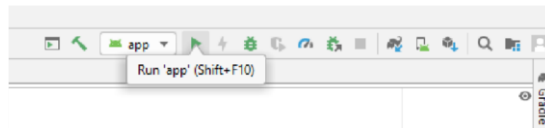
3. Configuración inicial del proyecto.
  - a. [Name]: Colocaremos el nombre de nuestra app. (Recuerde que este nombre será reflejado en el PlayStore al momento de publicarlo). Para este taller, usaremos **AMST[numeroGrupo]**.
  - b. [PackageName]: Paquete principal de código java, se obtiene automáticamente del nombre
  - c. [Save Location]: Dirección donde se ubica el proyecto en nuestra PC
  - d. [Language]: java
4. Seleccionamos **FINISH**



5. Como resultado se creará un proyecto vacío, solo presentado el mensaje “Hello World”

**Ejecutar nuestra app en el teléfono**

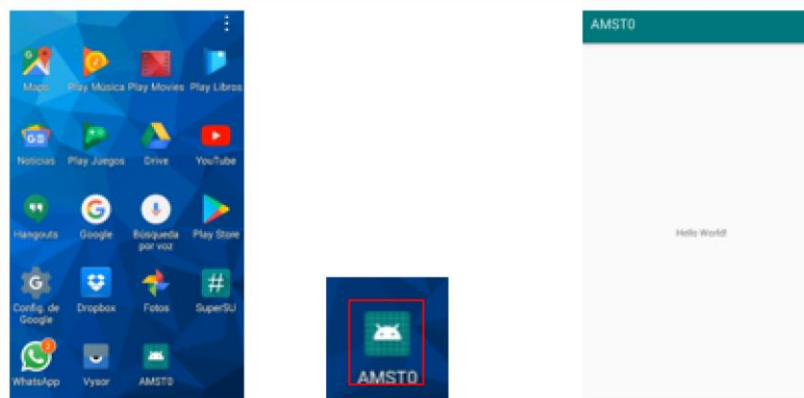
1. Del lado superior derecho de AndroidStudio aparecer la barra de ejecución ("RUN").
2. Buscar el icono play para ejecutar nuestra app (o usar el atajo Shift+F10).



3. Ahora aparecerá la ventana "Select Deployment target" (Seleccionar dispositivo a ejecutar). Ahora seleccionamos en "**Connected devices**" > [modelo del teléfono conectado]:

**Samsung SM G530**



Esto creara una aplicación local en nuestro celular, y podemos probarla en vivo.



4.

**TAREAS DE DESAFÍO:**

Una empresa dedicada a la venta de libros quiere mostrar su catálogo mediante una aplicación móvil (el modo de venta no se ha definido aún). Ha realizado un prototipo de baja resolución de lo que espera que tenga la aplicación.

1	2
 <p><i>Ilustración 1 Splash art con el logo de la empresa</i></p>	 <p><i>Ilustración 2 Pantalla de Inicio de sesión</i></p>
Una pantalla inicial donde solo se presente el logo de la empresa.	Seguido, se presenta una pantalla para realizar inicio de sesión.

3	4
 <p><i>Ilustración 3 Ventana principal o home</i></p>	 <p><i>Ilustración 4 Descripción del libro</i></p>
Una vez que se ingresó se procede a mostrar la página principal que cuenta con una lista de todos los libros con los que se cuenta y un cuadro de búsqueda por nombre de libro.	Cuando se selecciona un ítem de la lista se abrirá un cuadro de diálogo personalizado indicando la sinopsis del libro en cuestión.

5	6
 <p><i>Ilustración 5 Perfil de usuario</i></p>	 <p><i>Ilustración 6 Sección categorías de libros</i></p>
<p>Se cuenta con una sección perfil donde se detallan los datos del usuario.</p>	<p>En la sección de categorías se muestran las categorías de libros que existen en la empresa. Una vez que se seleccione una categoría, esta debe mostrar una lista de libros de solo esa categoría como la página de home.</p>

A usted y su equipo le han pedido que realice un prototipo de alta resolución para Android donde se implemente la idea del prototipo de baja.

#### Observaciones:

- La interfaz de usuario debe ser creativa e interactiva por cada equipo. No es necesario que sea una réplica del prototipo de baja resolución.
- La experiencia de usuario debe ser similar a la presentada: debe existir una forma de buscar libros, de categorizarlos, de presentar la sinopsis en un cuadro de dialogo personalizado y de hacer un login (inicio de sesion)
- No es necesario usar una base de datos interna, salvo que usted lo desee.
- No es necesario que ponga las mismas categorías de libros ni todas las existentes en el mercado. Con 3 o 4 categorías bastan. Así mismo con 3 o 4 libros por categoría es suficiente.

#### Entregables:

- El presente documento en Word con las actividades desarrolladas: preguntas de investigación, desarrollo de la aplicación donde se escribe las clases, activitys y archivos xml con una explicación corta acerca de lo que hace la activity (máximo de 3 líneas), captura de pantalla de las estadísticas de los commits realizados al proyecto (se encuentra en Insights>Contributors).
- Enlace del repositorio público de github, con el archivo ejecutable (.apk), y un video corto de ejecución de la aplicación.
- Los archivos (documento y apk) deben ser subidas por separado a Sidweb en la misma tarea no como un archivo comprimido.

#### RETO:

El equipo que presente el mejor prototipo de alta resolución puede recibir un punto extra. Se calificará en base a estos criterios:

- Interfaz de usuario
- Experiencia de usuario
- Código (buenas prácticas de programación)
- Elemento adicional que el equipo implemente con su prototipo de alta resolución.

## PREGUNTAS DE INVESTIGACIÓN:

### 1. ¿Qué otro tipo de servicios en línea (como GITHUB) existen?

#### Gitlab

Es un software gratuito de código libre creado en el año 2011 y que sirve como repositorio de git. Cumple funciones muy similares a GitHub como la creación y reparto de tareas para un proyecto o la gestión del código fuente utilizado, ya que nos provee una forma de visualizar el historial de modificaciones y cambios al código. (Pathak, 2021)

Adicionalmente posee mayores derechos para los usuarios puesto que nos brinda herramientas de ramificaciones protegidas en repositorios tanto públicos como privados. También contiene herramientas de integración continua que genera informes del proyecto en base a procesos de compilación y ejecución del mismo en ciertos intervalos de tiempo. (Anónimo, s.f.)

#### SourceForge

Es un sitio web fundado en el año 1999 que permite la colaboración conjunta para distintos proyectos de software publicados en la página. Actúa como un repositorio de software libre del cual varios usuarios pueden beneficiarse al utilizarlo dicho programa de forma gratuita, o pueden actuar como un contribuidor para mejorarlo. El sitio web lleva instancia de quien y que cambios se han hecho en el proyecto. Es compatible con Linux, Mac e IOs además de tener soporte para CVS,SSVN,Git y Mercurial (Carbajo, 2015)

#### Bitbucket

Es un servicio de alojamiento en la web con un sistema de control de versiones de Git y Mercurial orientado más a ámbitos profesionales o empresariales. Provee repositorios privados de forma gratuita además de un sistema de búsqueda semántica a través de la integración de Trello o Jira, dependiendo si se ha optado por la opción de gratuita del servidor, permitiendo el rastreo de errores de programación. (Anónimo, Facil Cloud, s.f.)

Su principal desventaja es que, a diferencia de GitHub, los repositorios son privados eliminando la posibilidad de integrar códigos de terceros que pudiesen facilitar el desarrollo de un proyecto. (Nuñez, 2016)

### 2. ¿Para qué sirve el archivo .gitignore y como se utiliza?

Le avisa al git de que archivos y directorios debe de ignorar en el proyecto del repositorio. Se puede usar esta carpeta debido a varios motivos, ya sea para garantizar aspectos de privacidad con ciertos códigos fuentes que no requieren ser subidos, para evitar que se creen carpetas secundarias creadas por herramientas de terceros o para reducir el tamaño final de nuestro proyecto. (Anónimo, Desarrollo web, 2016)

Su implementación es sencilla, solo se requiere crear una carpeta con el nombre “.gitignore” en la carpeta raíz, esta trabaja con texto plano por lo que solo es necesario escribir los archivos que no se requieran de la siguiente forma:

Si se desea ignorar cierta ruta, solo se coloca la dirección del archivo relativa al archivo “.gitignore” seguida del símbolo “/”.

También se puede utilizar el símbolo “\*” para ignorar archivos con ciertas terminaciones. Por ejemplo, al colocar dentro de nuestro archivo “\*.txt” le estaremos indicando al repositorio que no tome en cuenta ningún archivo de texto. (Sulbaran, 2021)

### 3. ¿Qué limitaciones tiene GITHUB?

Posee algunas limitaciones como que todos los repositorios gratuitos son públicos, por lo que si se quiere privacidad en un proyecto es necesario realizar un pago adicional. También posee limitaciones de espacios de 1 GB para el repositorio y de 100 MB para los archivos que contendrá. Por último, no brinda una documentación detallada de los procesos a seguir durante la importación y exportación de un proyecto ubicado en una plataforma distinta. (Clancy, 2018)

### 4. ¿Qué es una rama?

Se puede entender como un apuntador a una instancia específica de un proyecto a la hora de hacer un commit. Mientras la rama principal sigue desarrollándose, la nueva rama creada no cambiará y seguirá apuntando a dicha instancia especificada. (López, 2016)

Son espacios que nos permite trabajar en el mismo proyecto pero de forma independiente pudiendo tener accesos a los datos y archivos originales del repositorio sin afectar a la rama principal. (Vega, s.f.)

### 5. ¿Con cuál nombre, comúnmente, se le denomina a la rama principal?



Rama master (Anónimo, Git, s.f.)

## 6. ¿Cuál es el link de su repositorio?

[https://github.com/DiegoRonquillo4/PST\\_TA5\\_G2.git](https://github.com/DiegoRonquillo4/PST_TA5_G2.git)

## 7. ¿Cómo utilizo un repositorio público (utilizando el comando git clone)?

GitHub es un lugar de alojamiento de proyectos públicos de los que se pueden obtener accesos si se quiere realizar alguna aportación a este. Esto se puede hacer mediante el comando “git clone” que nos permite crear un repositorio local con una copia de casi todos los datos del servidor junto con las versiones de cada archivo y sus carpetas. (Anónimo, GitHub Docs, s.f.)

Para poder usar el comando “git clone” primero nos ubicamos en la página principal del repositorio que deseamos clonar, buscamos la opción que diga “clonar o descargar” y copiamos el URL del proyecto. Posteriormente abrimos una ventana terminal en nuestra computadora y nos ubicamos en la carpeta que va a contener el proyecto clonado, escribimos el comando “git clone” seguido del URL copiado y ejecutamos el comando.

Cabe recalcar que es necesario hacer instalaciones adicionales de ciertas dependencias de archivos del proyecto clonado, como npm para Node.js o composer para PHP, puesto que usualmente estos archivos son colocados en la carpeta “.gitignore”. (Anónimo, Git, s.f.)

### PREGUNTAS DE INVESTIGACIÓN

#### 1. ¿Qué utilidades tiene el manejo de ramas?

Poseen la ventaja de ser creadas de forma local y privada, brindando mayor seguridad al proyecto. Son de principal ayuda en trabajos de equipos numerosos puesto que permite una mejor repartición de tareas al brindar la oportunidad de trabajar con los archivos principales sin los riesgos de entorpecer el trabajo de otras personas o afectar en gran medida el código fuente original. (Anónimo, Backlog, s.f.)

Adicionalmente, el uso de ramas da la potestad de trabajar con funciones experimentales de desarrollo largo sin que frene otros avances y producción de la rama del código principal y, una vez finalizada dicha función se puede integrar al proyecto principal mediante la fusión de ramas. (Anónimo, Atlassian, s.f.)

#### 2. ¿Qué tipos de conflictos puede ocurrir durante el manejo y creación de ramas?

Un conflicto que puede surgir es que, con tantas ramas de trabajo distintas, el historial de actividad del proyecto se puede volver irreconocible, perdiendo así el hilo de desarrollo del trabajo. También se pueden apilar varias ramas abandonadas e inconclusas en el repositorio, consumiendo espacio de memoria de este. (Porto, 2018)

Por ultimo puede haber conflictos de modificaciones en donde una misma porción de un archivo ha sido cambiada desde dos ramas diferentes o si se ha eliminado dicho archivo mientras era utilizado en otra rama, provocando la aparición de errores dentro del repositorio. (Anónimo, Atlassian, s.f.)

#### 3. ¿Cuál es la diferencia entre checkout y checkout -b?

El comando “git checkout” opera sobre archivos, confirmaciones y ramas. Nos permite desplazarnos entre varias ramas preexistentes y se puede usar como un comando para deshacer los cambios en un archivo del directorio puesto que solo nos permite hacer el cambio de rama si nuestro directorio de trabajo está limpio, de lo contrario borrará cualquier alteración a la que no se le haya realizado un commit.

Por otro lado “git checkout -b” es un poco distinto puesto que se puede interpretar como un trabajo conjunto de las funciones “git branch” y “git checkout” ya que actúa como un método práctico para crear una nueva rama, con el nombre que se le indico al ejecutar el comando, y cambiarse instantáneamente a esta. (Carrillo, 2021)

#### 4. ¿Qué es la herramienta lint?

Es una herramienta de análisis estático de código. Tienen la funcionalidad de detectar aspectos sospechosos del código a nivel de arquitectura. Es un recurso bastante útil ofrecido por Android Studio que evita problemas estructurales del código mejorando en gran medida la eficiencia y confianza de nuestra aplicación, esto mediante informes descriptivos de los problemas detectados y su nivel de gravedad para el funcionamiento de la aplicación, ya sea utilización de variables antes de ser definidas, sobrepasar ciertos rangos numéricos o condiciones de verdad redundantes. (Anónimo, Developers, s.f.)

**5. Escriba al menos 5 buenas prácticas de estándares de programación para creación de aplicaciones móviles en Android Studio/Java.**

1. Crear un código con comentarios (documentación), variables entendibles, funciones claras, en general un código que sea fácil de entender para otros programadores.
2. Aligerar todo lo posible la aplicación, reduciendo los recursos que usa sin afectar la funcionalidad de la misma.
3. Tener en cuenta la depuración del código, de modo que se eliminen la mayor cantidad de errores que pueda tener la app. (Anónimo, Software Guru, s.f.)
4. Crear validaciones de todos los datos ingresados antes de que estos sean utilizados, mejorando la seguridad de la app ante posibles ataques.
5. Considerar cambios futuros que se realicen en la aplicación, ya sea a funciones, objetos, etc. (Anónimo, We live security, 2015)

**6. Investigue acerca de los tipos de aplicaciones móviles, su principal característica, las tecnologías con las que se programan y ejemplos de aplicaciones de cada tipo.**

**Aplicaciones móviles nativas**

Son aquellas diseñadas para un solo sistema operativo móvil. Para programar estas apps, se puede usar Kotlin para Android (Lenguaje de primer nivel declarado por Google) o Swift para iOS (Facilita los patrones de código y gestiona la memoria automáticamente)

Ejemplo: Pixelmator para iOS.

**Aplicaciones móviles híbridas (multiplataforma)**

Son aquellas capaces de funcionar en varios sistemas operativos a la vez.

Estas apps se pueden programar mediante Ionic (Un IDE que emplea estándares como HTML, CSS y JavaScript) o Xamarin (Que emplea código C# y XAML) (Bernal, 2021)

Ejemplo:

Snapchat, TikTok

**Aplicaciones móviles WEB**

Son aquellas que se desarrollan y son funcionales en varios sistemas operativos y se puede acceder a ellas mediante un navegador WEB.

La creación de estas apps emplea el uso de frameworks (PhoneGap, Flutter, React Native, entre muchos más). (Casans, 2020)

Ejemplo:

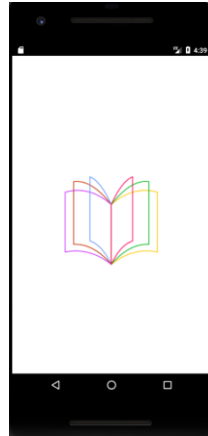
Facebook, Twitter, Netflix, Google

**DESARROLLO**

**Splash Screen**

[SplashActivity/ activity\\_splash.xml](#)

Actividad para mostrar la pantalla de carga (Splash Screen) de la aplicación. Para esto se utiliza un archivo logo.png.



```
public class SplashActivity extends AppCompatActivity {
    //SplashActivity para mostrar una pantalla Splash (pantalla de carga) al abrir la app.
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Luego se dirige al MainActivity (Pantalla de inicio de sesión)
        Intent intent = new Intent( packageContext: this, MainActivity.class);
        startActivity(intent);
        finish(); // evitar regresar a este activity
    }
}
```

### MyApp

Clase utilizada para definir la duración de la SplashScreen.

```
import ...
public class MyApp extends Application {
    //Clase para definir el tiempo del splash screen
    @Override
    public void onCreate() {
        super.onCreate();
        //Se define el tiempo del splashscreen en 3s.
        SystemClock.sleep( ms: 3000);
    }
}
```

### AdminSQLiteHelper

Clase heredada de SQLiteOpenHelper para administrar la base de datos SQLite. Dicha base de datos contendrá las tablas libros y usuarios. Además, se agregan registros por defecto a la tabla de libros.

### Inicio de Sesión

#### activity\_main.xml

Diseño xml que contiene dos EditText usuario y contraseña, un botón para iniciar sesión y una opción para registrarse. Además, cuenta con un ImageView de logo.png

#### MainActivity

Actividad para inicio de sesión. Valida si los datos ingresados se encuentran en la base de datos y concede paso a la aplicación o muestra un mensaje de credenciales inválidos.



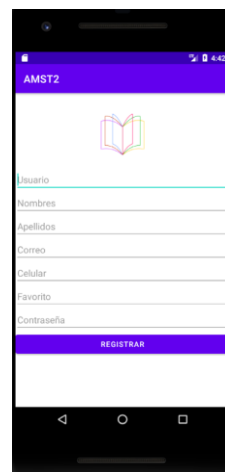
## Registro

### activity\_registrar.xml

Diseño xml que contiene 7 EditText con los campos necesarios para realizar un insert a la base de datos. Un botón para registrar los datos y un ImageView con la imagen logo.png.

### RegistrarActivity

Actividad para registrar un nuevo usuario. Al pulsar el botón registrar, se valida que los campos estén llenos. Si los campos se encuentran completos, se procede a registrar la información y se vuelve a la MainActivity.



## BottomNavigationView

### HomeActivity

Actividad para definir el BottomNavigationView de la aplicación. Contiene 3 fragments: Home, Category y Profile. Se utilizan los archivos bottom\_menu.xml y my\_nav.xml para definir botones y navegación.

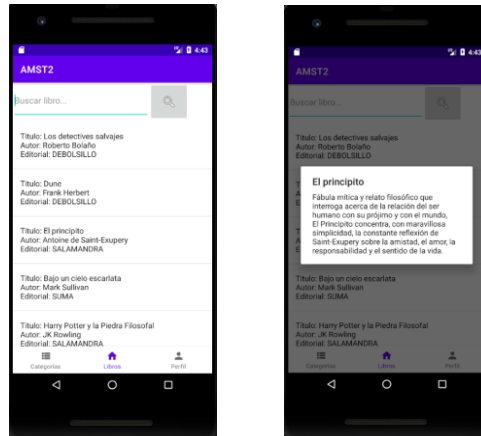


### Fragment\_home.xml

Diseño xml que contiene un EditText, un botón para buscar y un ListView.

### HomeFragment

Fragmento que presenta la lista de libros de la base de datos en una ListView, mostrando título, autor y editorial. Cada elemento de la lista tiene un cuadro de diálogo con la descripción del libro al hacer click en él. Además, presenta una opción para buscar por título y mostrar en la ListView.



fragment\_Profile.xml

Diseño xml que contiene varios TextViews para mostrar toda la información del usuario con sesión iniciada. ProfileFragment

Fragmento que presenta la información del usuario en texto (usuario, nombres, apellidos, correo, celular, favorito). Esta información fue enviada a través de un Intent desde el MainActivity.



fragment\_category.xml

Diseño xml que contiene 3 botones de categorías de libro. Cada botón tiene asignada su respectiva acción onClick en el archivo Java.

CategoryFragment

Actividad que permite elegir filtrar por categoría de libro. Se inicia la actividad CategoryActivity enviando información de la categoría seleccionada por medio de un intent.



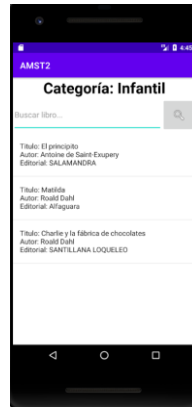
Activity\_category.xml

Diseño xml similar al de fragment\_home.xml. Una ListView acompañada de una opción para buscar (EditText más botón).

CategoryActivity

Actividad que según la categoría seleccionada muestra una ListView con la lista de libros de dicha categoría. Cada uno con su cuadro de diálogo con descripción. También presenta la opción de buscar por título tomando en consideración la categoría.





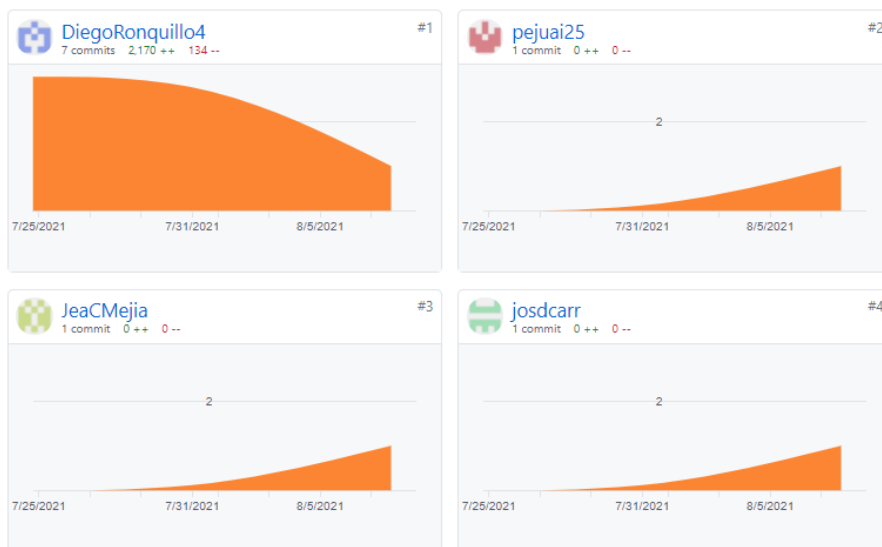
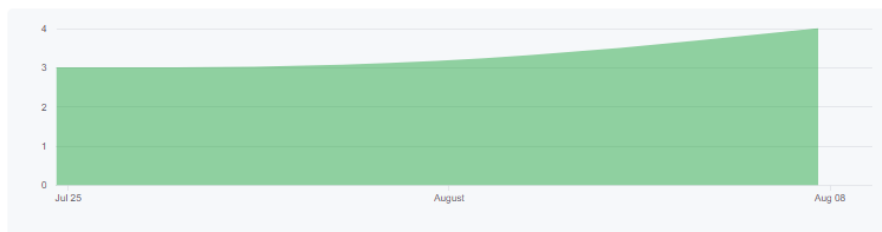
LINK VÍDEO: <https://youtu.be/kAkhykx6WEg>

## ESTADÍSTICAS

Jul 25, 2021 – Aug 9, 2021

Contributions: Commits ▾

Contributions to master, excluding merge commits and bot accounts



REPOSITORIO GITHUB: [https://github.com/DiegoRonquillo4/PST\\_TA5\\_G2](https://github.com/DiegoRonquillo4/PST_TA5_G2)