PERÍODO 2019

TRABAJO AUTÓNOMO 2

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

ELABORADO POR: ADRIANA COLLAGUAZO JARAMILLO

ITINERARIO: APLICACIONES MÓVILES Y SISTEMAS TELEMÁTICOS

CARRERA DE INGENIERÍA EN TELEMÁTICA

FIEC-ESPOL

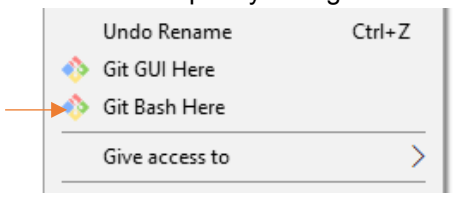
Trabajo Autónomo:
Objetivo de Aprendizaje: Desarrollar aplicaciones móviles sencillas considerando las características de la programación de dispositivos móviles.
Recursos: Android Studio, GIT (software). Github (online).
Duración: 5 horas

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

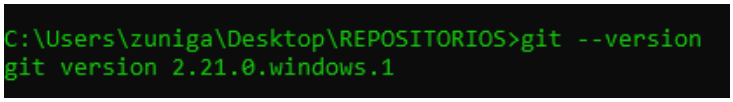
ACTIVIDADES
Paso 1: Crear un repositorio.

1. Creamos un proyecto de Android Studio, el cual vamos a alojar en nuestro repositorio.

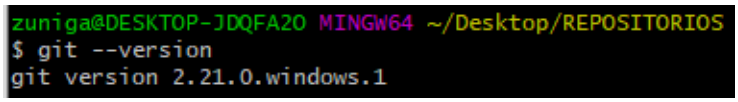
2. Dentro de la carpeta del proyecto, abra la línea de comandos de GIT (GIT CLI). Podemos encontrarlo, dando clic derecho dentro de la carpeta y escogemos la opción “GIT BASH HERE”.



   - *En caso de que no se disponga de GIT CLI, también se puede utilizar CMD de Windows/Ubuntu. Para pobar que GIT ha sido instalado correctamente, utilice el comando “git --version”.*

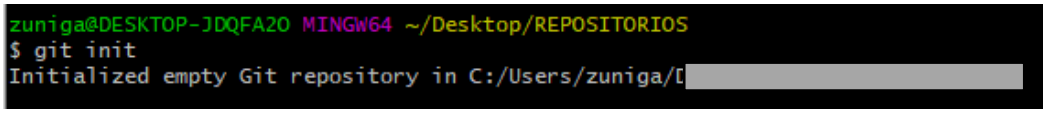


GIT en línea de comandos de Windows



GIT BASH propia

3. Para crear un nuevo repositorio, utilice el siguiente comando “git init”.



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

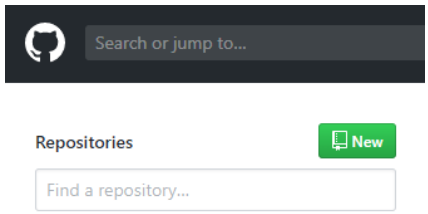
4. Agregamos todos los archivos del proyecto a nuestro **repositorio local** con el comando: “git add --all”.

```
zuniga@DESKTOP-JDQFA2O MINGW64 ~/Desktop/AMST_3 (master)
$ git add --all
warning: LF will be replaced by CRLF in gradlew.
The file will have its original line endings in your working directory
```

5. 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.

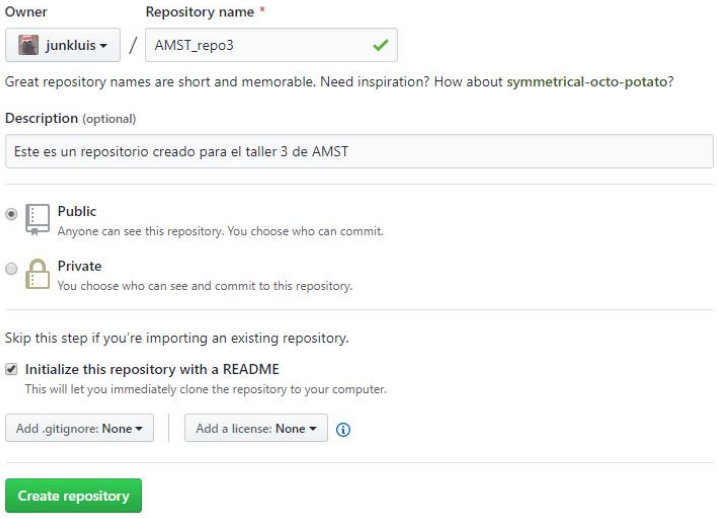
```
zuniga@DESKTOP-JDQFA2O MINGW64 ~/Desktop/AMST_3 (master)
$ git commit -m "Commit inicial"
[master (root-commit) 9430617] Commit inicial
 37 files changed, 646 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .idea/encodings.xml
 create mode 100644 .idea/gradle.xml
```

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

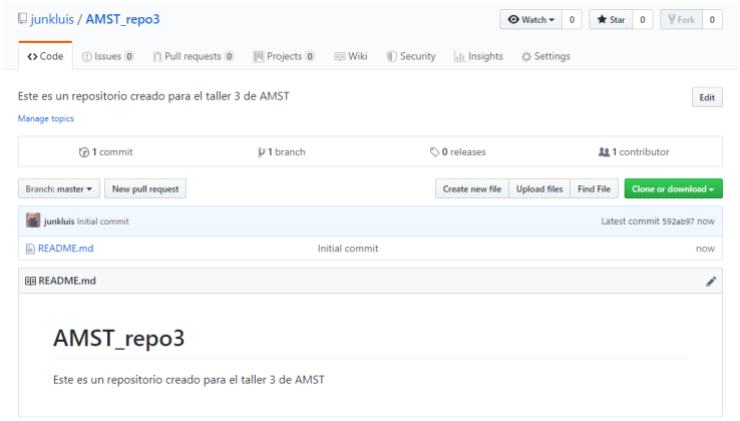


7. La información requerida para crear un repositorio, se muestra a continuación:

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

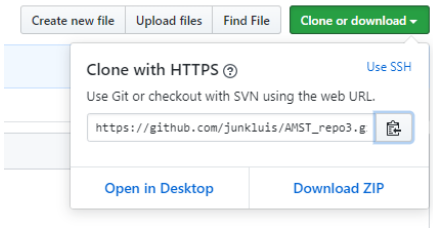


8. Una vez ingresados todos los campos, damos clic en “Create repository”.



Vista de mi repositorio vacío.

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



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

```
zuniga@DESKTOP-JDQFA2O MINGW64 ~/Desktop/AMST_3 (master)
$ git remote add origin https://github.com/junkluis/AMST_repo3.git
```

11. 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)]

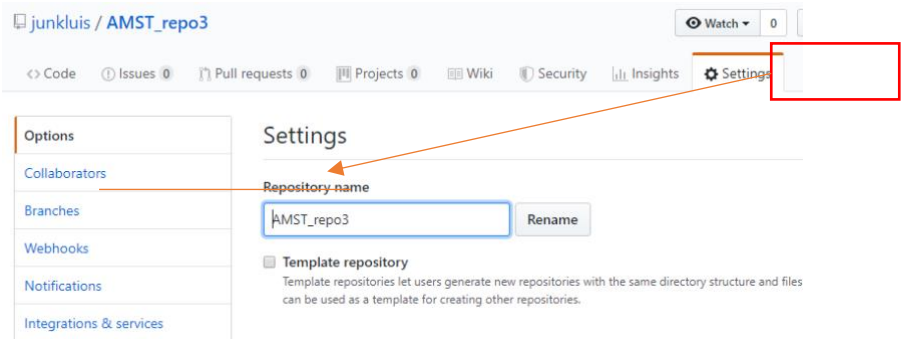
12. Repetir los pasos 4, 5 para actualizar los cambios realizados y el comando git push origin master.

Preguntas de investigación:

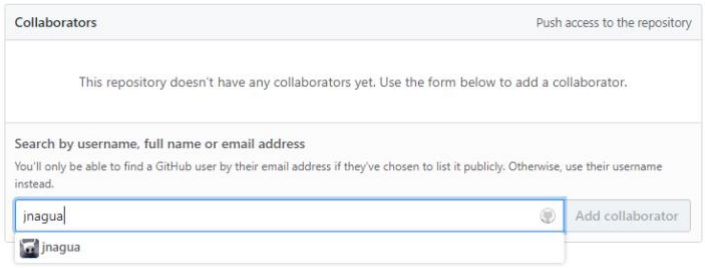
1. ¿Qué otro tipo de servicios en línea (como GITHUB) existen?
2. ¿Para qué sirve el archivo. gitignore y como se utiliza?
3. ¿Qué limitaciones tiene GITHUB?
4. ¿Qué es una rama?
5. ¿Cuál es el link de su repositorio?
6. ¿Cómo utilizo un repositorio público (utilizando el comando git clone)?

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

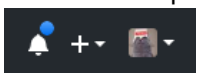
1. Para habilitar la modificación a otros miembros de mi grupo, debemos darle acceso. [Incluso si el proyecto es libre, solo pueden modificarlo quienes han sido invitados]. Para esto de clic en el tab “Settings/Configuración” > Collaborators / Colaboradores



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



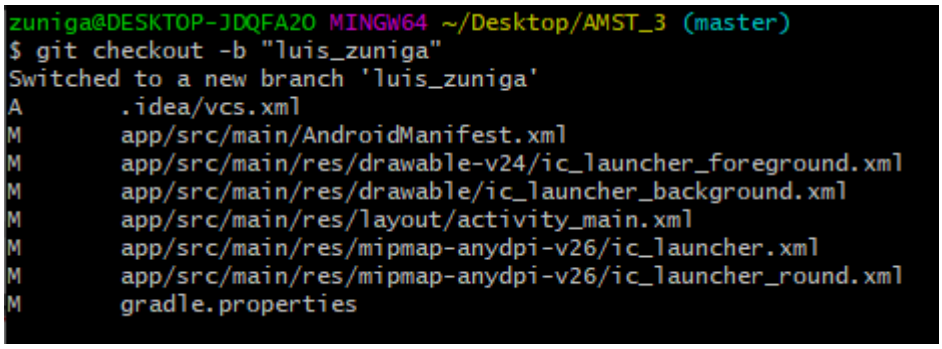
3. Una vez agregados, es necesario aceptar las invitaciones para poder realizar PUSH (cambios al repositorio). Puede revisar las invitaciones en la campana a lado del usuario.



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”



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.

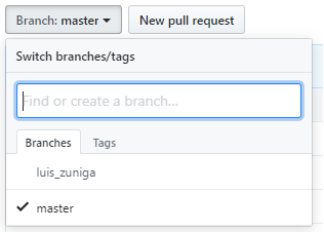
Git add --all	Agrega todos los cambios a nuestra rama.
Git commit -m “cambios a mi rama”	Agrega un commit en mi rama, indicando los cambios que realice.
Git push origin [nombre rama]	Subimos los cambios al repositorio (pero solo dentro de la página).

```
zuniga@DESKTOP-JDQFA2O MINGW64 ~/Desktop/AMST_3 (luis_zuniga)
$ git add --all

zuniga@DESKTOP-JDQFA2O 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-JDQFA2O 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).

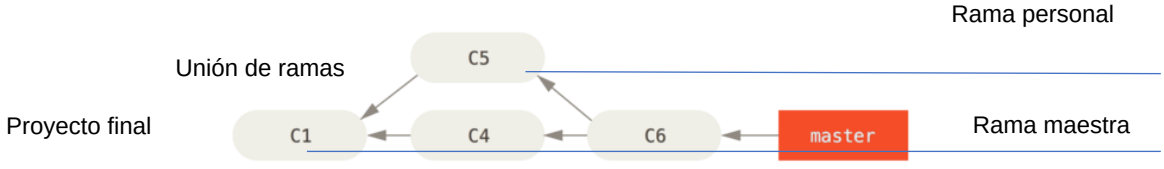


Preguntas de investigación

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 **chekout** y **checkout -b?**

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

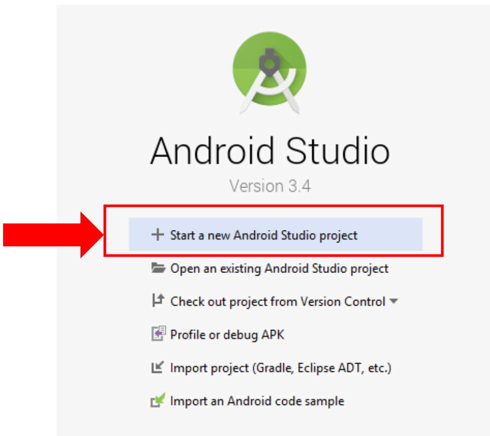
3. Traemos los cambios realizados en la rama única.
   a. Git merge [nombre rama]

```
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
```

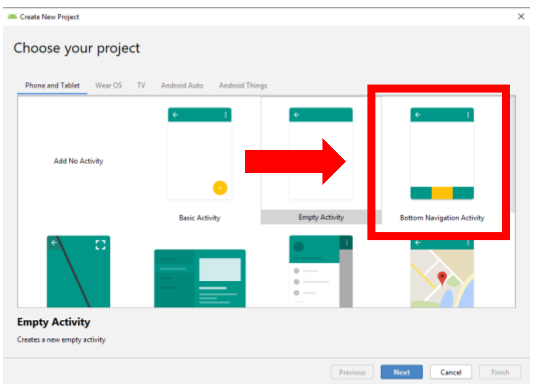
Nota: Esto indica los archivos que han sido modificados.

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".



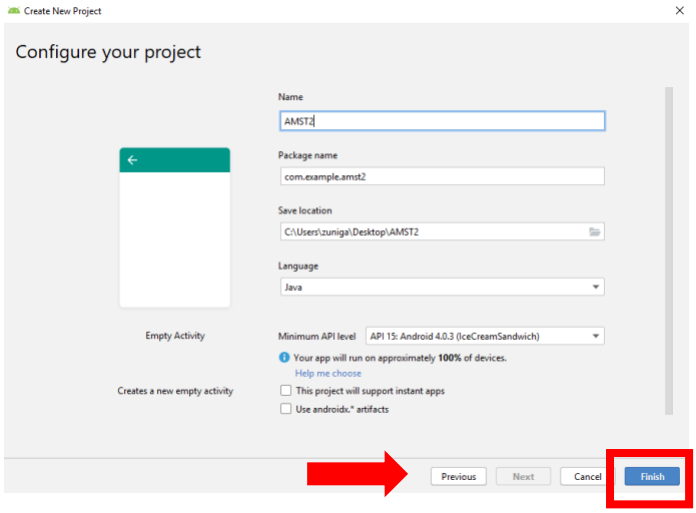
2. Seleccionar el tipo de proyecto: Para esta práctica escogeremos la pestaña **Phone and Tablet > Empty Activity**. Otro tipo de actividades viene por defecto con componentes no necesarios para este taller.



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]: Direccion donde se ubica el proyecto en nuestra PC
d. [Language]: java

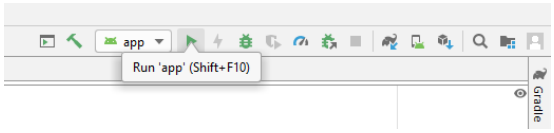
4. Seleccionamos **FINISH**



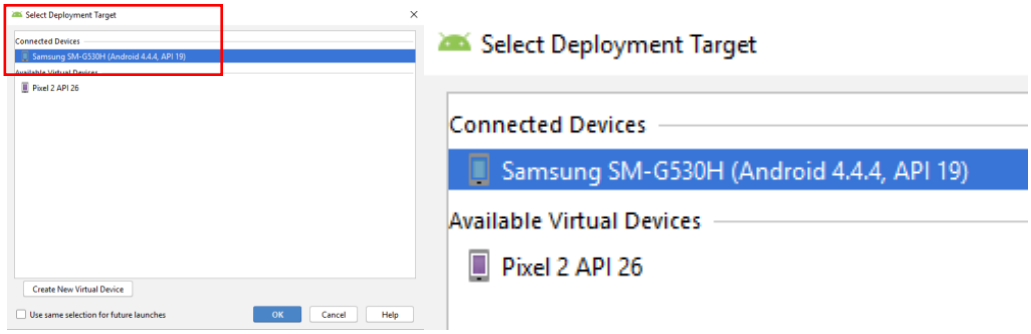
Como resultado se creará un proyecto vacío, solo presentado el mensaje “Hello World”

Ejecutar nuestra app en el teléfono

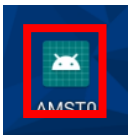
1. Del lado superior derecho de AndroidStudio aparecer la barra de ejecución (“RUN”).
2. Buscar el icono play para ejecutar nuestra app (o usar el atajo Shift+F10).

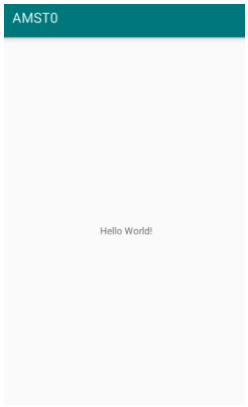


3. Ahora aparecerá la ventana “Select Deployment target” (Seleccionar dispositivo a ejecutar). Ahora seleccionamos en **“Connected devices” > [modelo del teléfono conectado]: Samsung SM G530**



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





TAREAS DE DESAFÍO:

1. Cree un repositorio de su grupo, cree un menú donde cada integrante usara un componente diferente. Componentes por usar:
   - Video view: Obtener un video de youtube.
   - Calendar view: Mostrar el calendario con un tarea.
   - Mostrar un mapa con Google maps.
   - Mostrar un gráfico lineal estático.
   - Crear un menú usando un botón flotante.

FORMATO DEL TRABAJO

El trabajo autónomo será desarrollado en el siguiente formato:

- Nombre del archivo: AMST_Trabajo Autónomo A_Grupo B_Apellido1_Apellido2_Apellido3
- *(*) Siendo A el número del trabajo y B el número del grupo*
- Nombre de la materia y paralelo 1
- Título del trabajo: Ejemplo: Trabajo Autónomo A - Tema
- Nombre de la profesora
- Número de grupo
- Nombres/Apellidos de los integrantes del grupo que hayan desarrollado el trabajo
- Fecha de inicio y fin del trabajo
- Resultados de las actividades planteadas: Explicación de las actividades ejecutadas, incluyendo las imágenes del proceso.
- Conclusiones y Recomendaciones: Respecto a lo aprendido durante el desarrollo del trabajo.
- Referencias bibliográficas: Colocar los documentos, enlaces web o libros consultados.