MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Unidad 1

Scrum y preparación del entorno

Universidad Tecnológica de Pereira

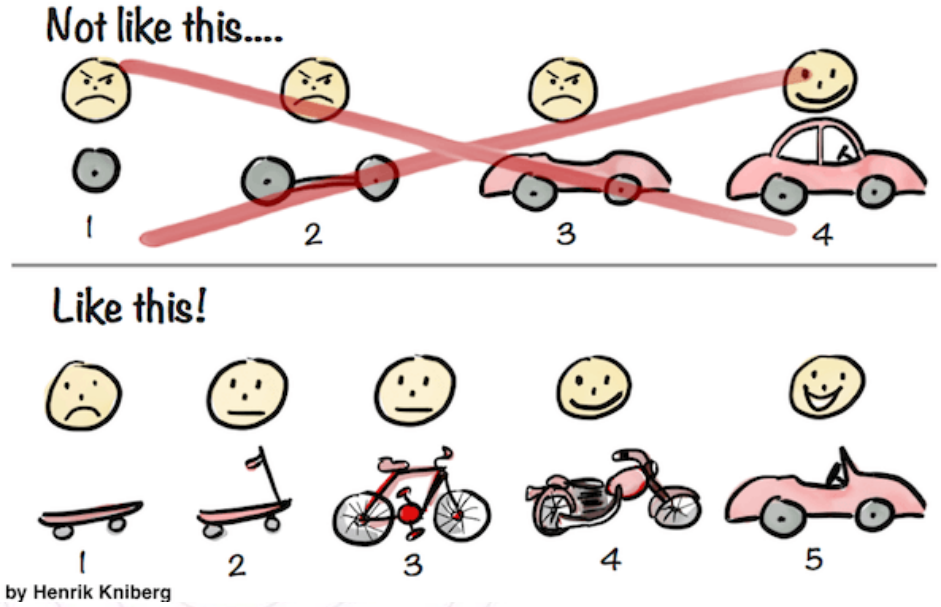
MisionTIC2022

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

1. SCRUM

Scrum

Es una metodología ágil y flexible que se utiliza para gestionar proyectos, su principal objetivo es apoyar proyectos donde el entorno de estos es muy cambiante en cuanto a tecnologías, requerimientos y equipos, maximizando el retorno de la inversión (ROI) al cliente.



Aprende

SCRUM

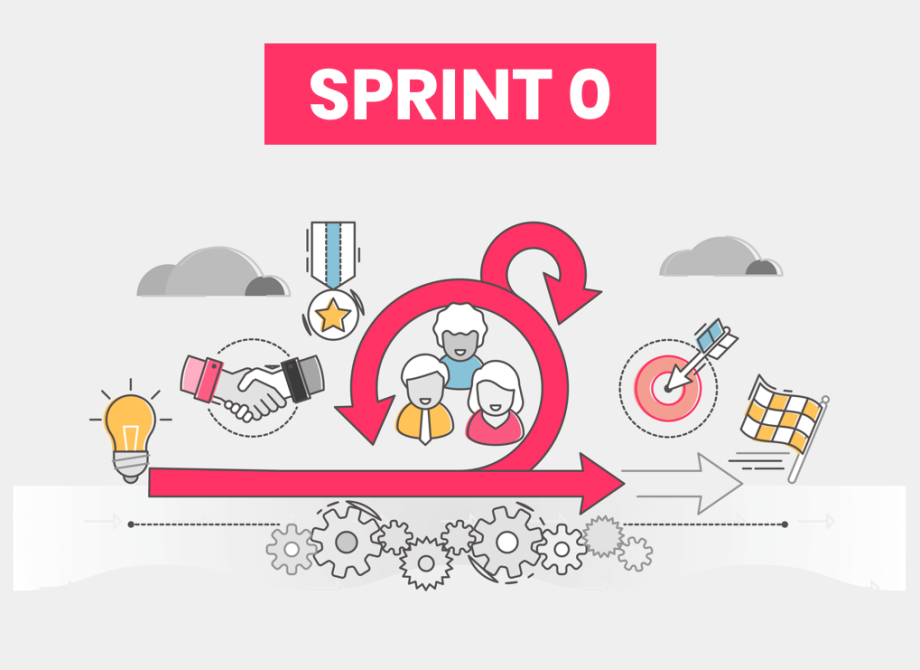
EN 10 MINUTOS

Scrum

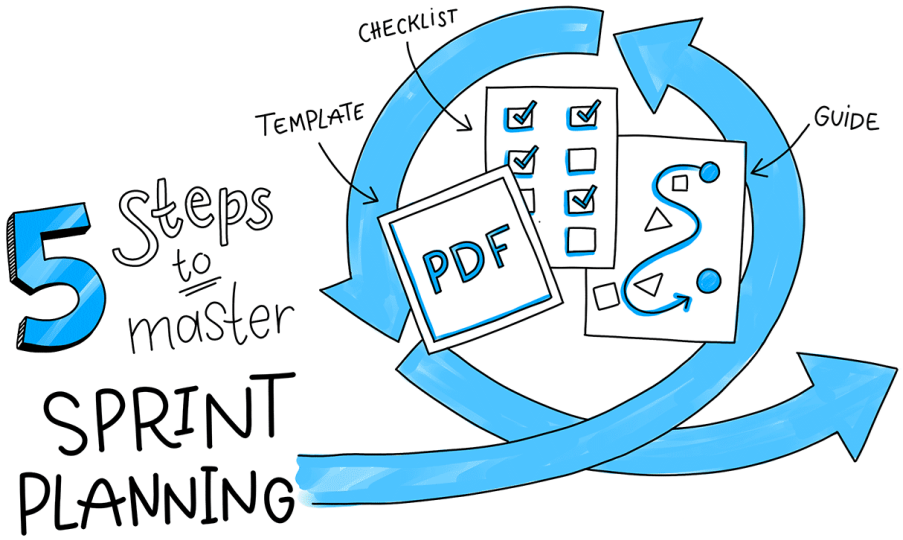
Utilizando Scrum:

- Se mejora la calidad del producto final.
- Se mejora la productividad de los integrantes.
- Se realiza un seguimiento diario de los avances del proyecto
- Se realizan iteraciones cortas.
- Se realizan sprints, con entregables semi-cerrados que sirven para que el cliente pueda obtener los beneficios del proyecto en forma incremental.
- Los entregables deben ser presentados como un resultado finalizado, lo que se conoce como incremento de producto.

Esto le permite ver al cliente el progreso y resultado del proyecto en varias etapas dentro del mismo proyecto.



El proceso de Scrum

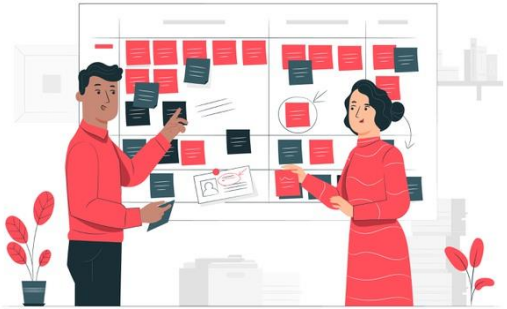


Se trabaja con iteraciones cortas **(Sprints)**, donde cada iteración (normalmente de 2 semanas) debe representar **un incremento de producto** que pueda ser presentado al cliente como un resultado finalizado.

planio

El producto a desarrollar

Las características a desarrollar durante cada iteración salen del **Product Backlog**: Nombre con el que se conoce al conjunto de requisitos priorizados.



Estas características se determinan al inicio de cada sprint, en la reunión que se conoce como **Sprint Planning**.

Sprint Planning

En esta reunión el **Product Owner** en conjunto con el equipo (**SCRUM Team**) determinan que elementos del **Product Backlog** se pueden comprometer a desarrollar y se clarifican las historias seleccionadas.

Esta nueva lista de tareas a realizar se conoce como **Sprint Backlog**. Una vez comenzado el sprint, tanto las historias que conforman el Sprint Backlog como los requerimientos de las mismas no pueden ser modificados.

Ejecución del sprint en Scrum

Se realizan reuniones diarias (**Daily Scrum**) durante el sprint con el fin de mantener la comunicación y la productividad del equipo, normalmente frente a un tablero y con el uso de POST ITS.

En estas reuniones cada integrante participa respondiendo a 3 preguntas:

- ¿En que trabajó el último día?
- ¿En que va a trabajar hoy?
- ¿Hay alguna tarea que se vea bloqueada por algún motivo?

Es el **Scrum Master** el que debe asegurar que el equipo pueda cumplir con su trabajo.

Ejecución del sprint en Scrum

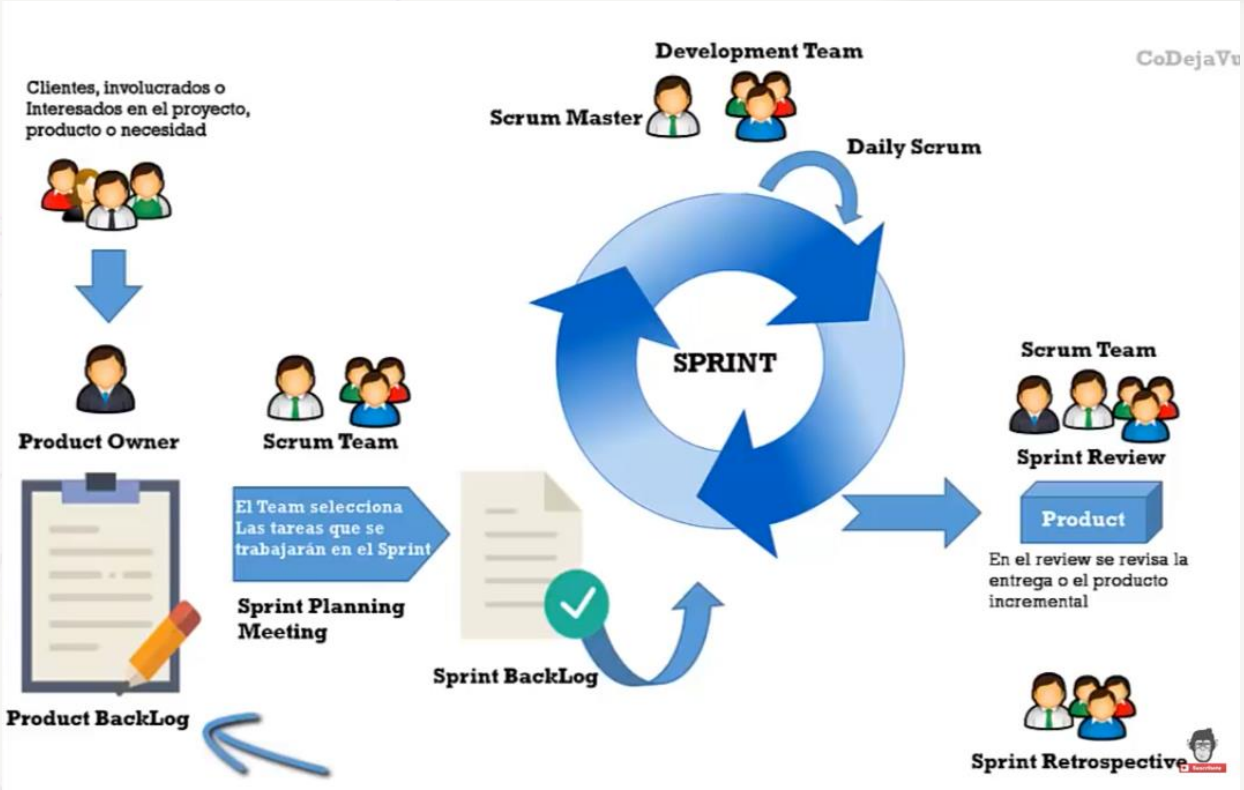
Una vez finalizado el sprint se realizan 2 reuniones más:

Sprint Review

Aquí analiza el trabajo realizado, que ítems fueron terminados y cuáles no y se presenta al cliente el trabajo finalizado.

Sprint Retrospective

Aquí los integrantes del equipo dan su opinión sobre el sprint finalizado y se proponen mejoras sobre el proceso.



Resumen SCRUM

Universidad Tecnológica de Pereira

MisionTIC2022

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

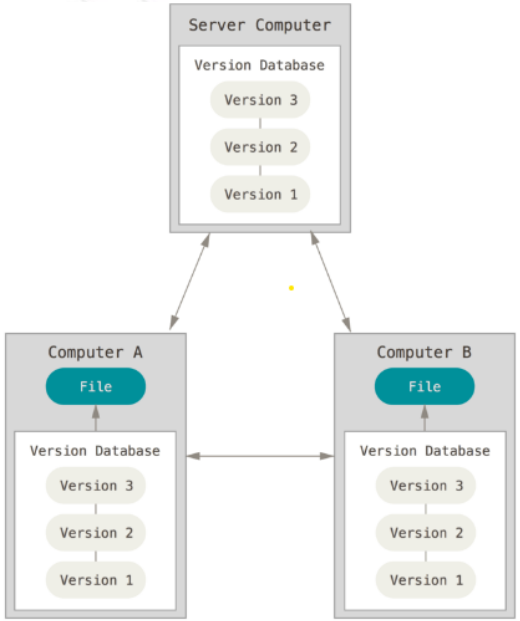
2. Git y Github

Git - Funcionamiento básico

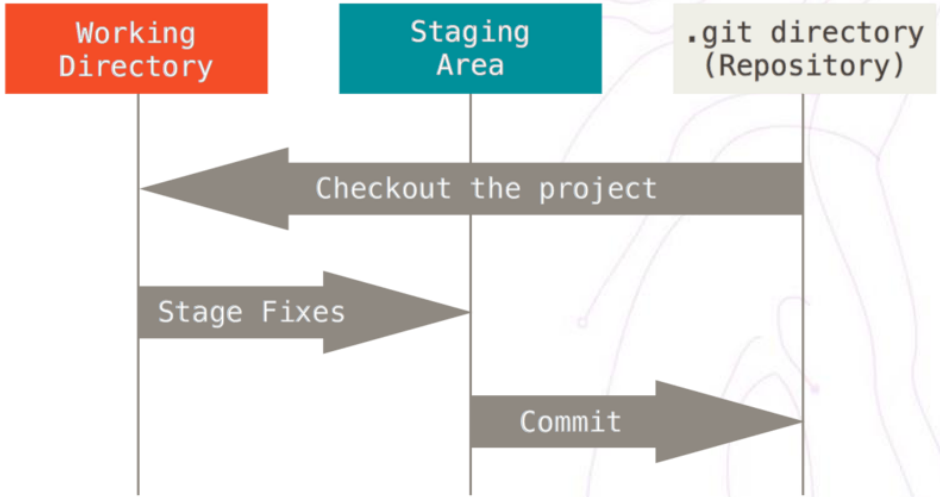
¿Por qué un Sistema de Control de Versiones?

- No trabajamos solos.
- Si modificamos un proyecto directamente, no tenemos constancia de cómo era antes de los cambios.
- Seguridad.

Estructura de Git



Los tres estados de Git



Working Directory: Traemos una copia de los ficheros del repositorio: checkout (sobreescribimos) o pull (merge).

Staging Area: Preparas los archivos, añadiendolos a tu área de preparación : add.

.git directory: Confirmar los cambios tal y como están en el área de preparación y guarda un snapshot : commit

Trabajo en equipo

- Cada equipo tendrá un repositorio Github
- Trabajar en el repositorio
  a. Inicializar el repositorio local: **git init**
  b. Traer el directorio remoto: **git clone <repositorio>**
  c. Crear rama para la funcionalidad a trabajar: **git checkout -b <funcionalidad>**
  d. Añadir los cambios necesarios a la rama.
  e. Añadir los cambios al stage: **git add .**
  f. Registrar los cambios: **git commit –m "<mensaje>"**
  g. Subir cambios al repositorio remoto: **git push -u origin <funcionalidad>**
  h. Crear un Pull Request (PR)
     - Hacer revisión del código
     - Realizar mezcla del PR

Trabajando con ramas en Git

- Crear una rama:

```
$ git checkout -b rama1
```

- Ver en que rama estamos

```
$ git branch
```

- Cambiar de rama

```
$ git checkout master
```

- Ver los cambios entre ramas

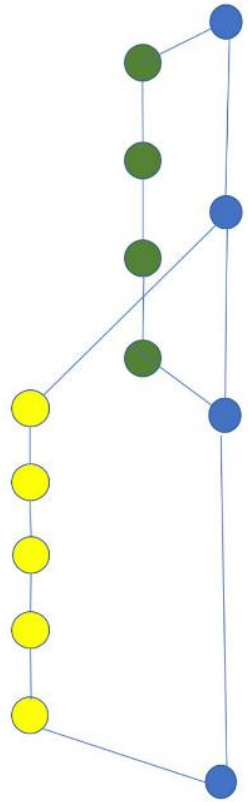
```
$ git diff --stat master rama1
```

- Fusionar ramas

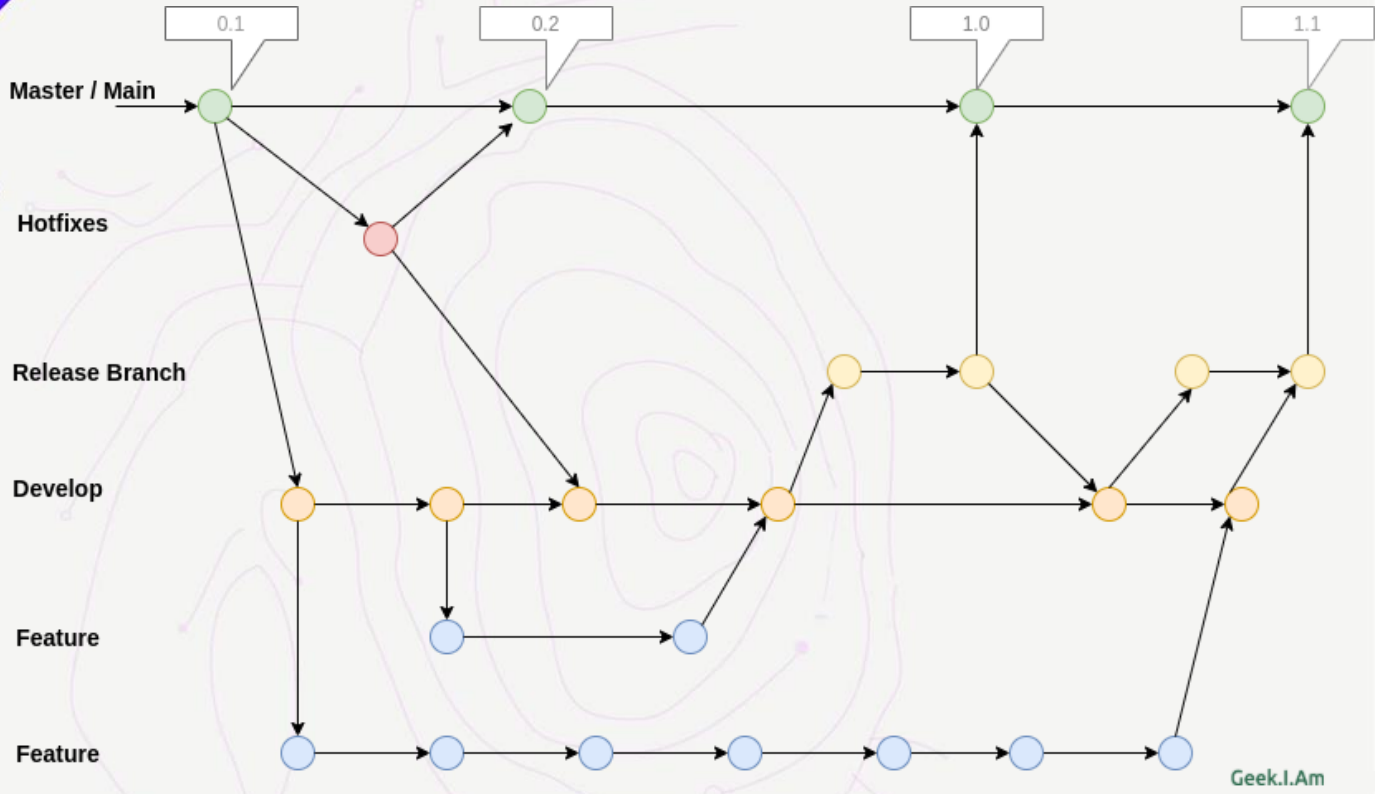
```
$ git checkout master
$ git merge --no-ff rama1
```

- Eliminar la rama

```
$ git branch -d rama1
```



Git Flow


0.1
0.2
1.0
1.1
Master / Main
Hotfixes
Release Branch
Develop
Feature
Feature
Geek.I.Am

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Git Flow

¿CÓMO TRABAJAR EN EQUIPO CON GIT?

RAMA MASTER **(No puedes trabajar aquí)**: Es donde está el proyecto principal, nunca debes tocarla.

RAMA DEV

(Aquí es donde puedes trabajar): Saca las ramas que quieras y trabaja sin preocupaciones.

1. Cada desarrollador puede sacar su propia rama de la "rama dev" y trabajar en lo que le corresponde.

2. Luego estas ramas se vuelven a unir a través de Merge.

Y Master está actualizado con todos los cambios hechos por los programadores y la app funcionando.

3. Si hay conflictos se deben corregir (Líder de proyecto o integración continua).

3. Se integran los cambios a la rama Master.

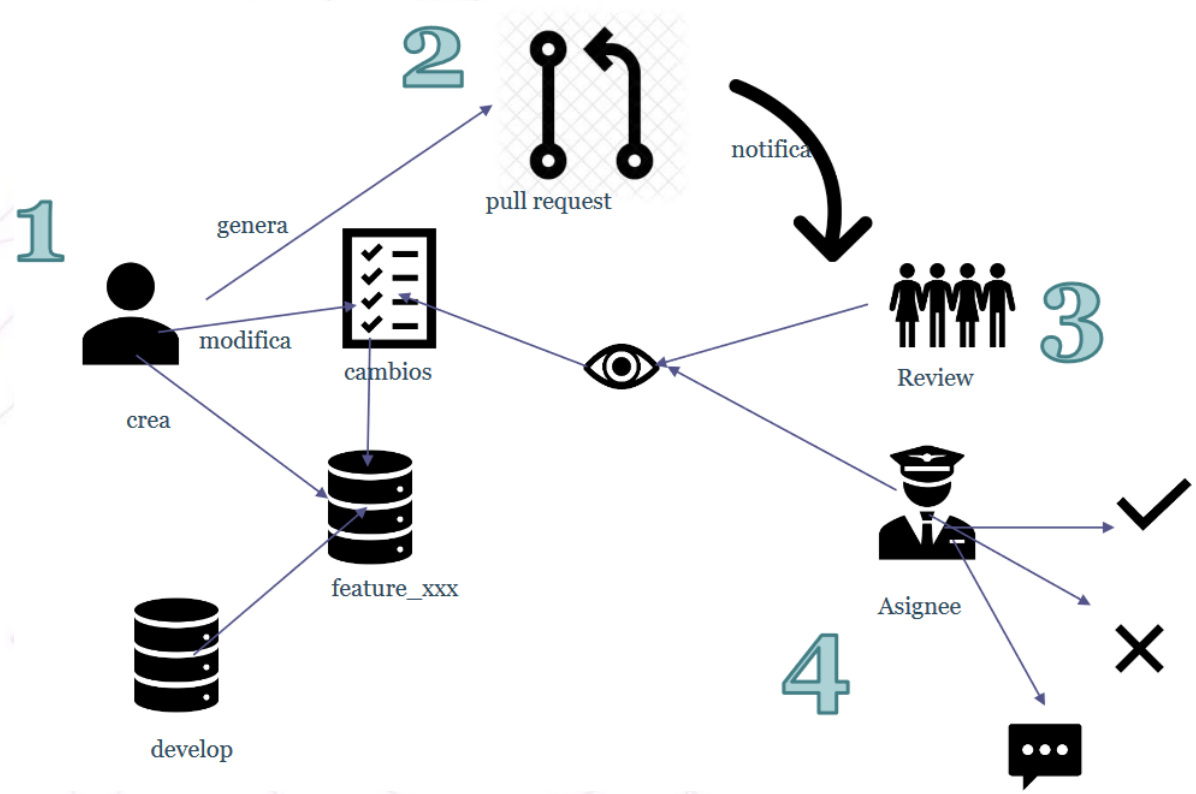
Aprende cómo usan Git los equipos de desarrollo de software en:

ed.team/cursos/git-workflow

EDteam

Pull request

- Crear la rama
- Subir los cambios en local
- Subir los cambios
- Ir a github y solicitar una pull request
- Añadir los comentarios: (también se pueden añadir revisores)
- Los revisores de código tienen tres opciones:
  - añadir comentarios
  - aceptar los cambios
  - rechazarlos



Universidad Tecnológica de Pereira

MisionTIC 2022

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

3. Flutter

¿Qué es Flutter?

Flutter es un conjunto de herramientas de Google para construir experiencias nativas de alta calidad para móvil, web y escritorio en tiempo récord.

Flutter

MacOS y Windows: https://www.youtube.com/playlist?list=PLCKuOXG0bPi3xBRYOmcfoqrchgRJOafo9

Requerimientos Mínimos recomendados

Windows

- **Sistema Operativo**: Windows 10 o superior
- **Disco**: 1.64 GB disponibles
- **Herramientas**:
  - Windows PowerShell 5 o superior
  - Git for Windows 2.x

macOS

- **Sistema Operativo**: macOS
- **Disco**: 2.8 GB disponibles
- **Herramientas**:
  - git 2.x

Linux

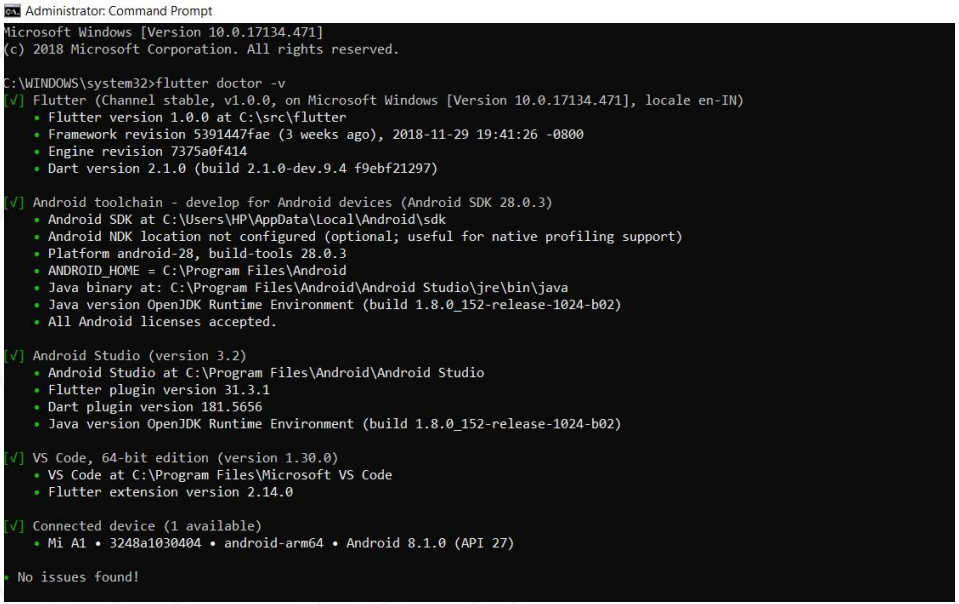
- **Sistema Operativo**: Linux (64-bit)
- **Disco**: 600 MB disponibles
- **Herramientas**:
  - `bash`
  - `curl`
  - `file`
  - `git 2.x`
  - `mkdir`
  - `rm`
  - `unzip`
  - `which`
  - `xz.utils`
  - `zip`

Instalar Flutter (https://docs.flutter.dev/get-started/install)

1. Descarga el siguiente paquete de instalación para obtener la versión más reciente stable release del SDK Flutter: (flutter_windows_3.3.3-stable.zip)
2. Extraiga el archivo zip y coloque el contenido de flutter en la ubicación deseada de instalación para el Flutter SDK (ej. `C:\src\flutter`; no instale flutter en un directorio como `C:\Program Files\` que requiere permisos de administrador).
3. Actualizar el path
   a. Desde la barra de busqueda en Inicio, escribe 'env' y selecciona **Editar variables de entorno para tu cuenta**
   b. Debajo de **Variables de usuario** verifica si existe una entrada llamada **Path**:
      i. Si la entrada existe, agrega la ruta completa a `flutter\bin` usando ; como separador de los valores existentes.
      ii. Si la entrada no existe, crea una nueva variable de usuario llamada Path con la ruta completa `flutter\bin` como su valor.

Instalar Flutter (https://docs.flutter.dev/get-started/install)

4. Ejecutar `flutter doctor`
5. Instalar Android Studio
    a. Descarga e instala Android Studio.
    b. Inicia Android Studio, y sigue todo el '**Android Studio Setup Wizard**'. Este Instalará la versión más reciente de Android SDK, Android SDK Platform-Tools y Android SDK Build-Tools, Las cuales son requeridas por Flutter cuando se desarrolla para Android.
6. Configurar tu dispositivo Android
7. Configurar el emulador de Android
8. Configurar Android Studio
9. Configurar Visual Studio Code (opcional)
10. Ejecutar `flutter doctor -v`



Configurar tu dispositivo Android

Conecta tu dispositivo Android a tu computadora con un cable USB. El tipo de conexión correcta debería aparecer como connected as a media device (conectado como un dispositivo de medios).

También necesitarás tener habilitadas las opciones de desarrollador en tu dispositivo Android. Hecho, sigue estos pasos (esto funcionará en la mayoría de los dispositivos Android):

Paso 1

Abre el menú **Settings** (Configuraciones) en tu teléfono Android y desplázate hasta la parte inferior.

Paso 2

Pulsa en **About phone** (Acerca del teléfono) y desplázate hacia abajo de nuevo hasta que veas la opción **Build number** (Número de compilación).

Paso 3

Pulsa en el número de compilación varias veces. Pronto deberías ver una ventana emergente que dice algo similar a **You are five taps away from being a developer** (Estás a cinco pulsaciones de ser un desarrollador).

Paso 4

Sigue pulsando hasta que la ventana emergente diga que eres un desarrollador.

Paso 5

Regresa al menú principal **Settings > System > Advanced** (Configuraciones > Sistema > Avanzado). Las opciones de desarrollador deberían ser la penúltima opción. Activa las opciones de desarrollador.

Paso 6

En Android Studio navega hasta el menú superior y selecciona **Run 'app'** (Ejecutar 'aplicación'). Android Studio mostrará un cuadro de diálogo en el que puedes elegir en qué dispositivo ejecutar tu aplicación Android. Elige el dispositivo que conectaste y haz clic en el botón **OK**

Configurar el emulador de Android

- Simula dispositivos Android en una computadora para que puedas probar tu app en diferentes dispositivos y niveles de API de Android sin necesidad de contar con los dispositivos físicos.
- Proporciona casi todas las funciones de un dispositivo Android real.
- Puedes simular llamadas y mensajes de texto entrantes, especificar la ubicación del dispositivo, utilizar diferentes velocidades de red, probar sensores de rotación y otros sensores de hardware, acceder a Google Play Store y mucho más.

Requisitos y recomendaciones

El uso de la aceleración de hardware implica requisitos adicionales en Windows y Linux:

- **Procesador Intel en Windows o Linux**: procesador Intel compatible con Intel VT-x, Intel EM64T (Intel 64 Bit) y funcionalidad Execute Disable (XD)
- **Procesador AMD en Linux**: procesador AMD compatible con AMD Virtualization (AMD-V) y extensiones complementarias de transmisión SIMD 3 (SSSE3)
- **Procesador AMD en Windows**: Android Studio 3.2 o posterior, y Windows 10 lanzado después de abril de 2018 para Windows Hypervisor Platform (WHPX)

Imagen tomada de:https://www.freepik.es/

Universidad Tecnológica de Pereira

Mision TIC 2022

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Dispositivos virtuales de Android (AVD)

Configurar Android Studio

1. Inicie Android Studio.
2. Abra preferencias de complementos (**Preferences>Plugins** en macOS, **File>Settings>Plugins** en Windows & Linux).
3. Seleccione **Browse repositories…**, elige el complemento de **Flutter** y presione `install`.
4. Presione `Yes` cuando aparezca para instalar el complemento de **Dart**.
5. Presione `Restart` cuando aparezca.

Configurar Visual Studio Code (opcional)

1. Inicie VS Code
2. Llame **View>Command Palette…**
3. Teclee `install`, y seleccione la acción **Extensions: Install Extension**
4. Introduzca `flutter` en el campo de búsqueda, seleccione **Flutter** en la lista, y presione Install
5. Llame a **View>Command Palette…**
6. Teclee `doctor`, y seleccione la acción **Flutter: Run Flutter Doctor**
7. Revise la salida en el panel **OUTPUT** para cualquier inconveniente.