**Universidad de las Américas**

**Facultad de Ingeniería y Ciencias Aplicadas**

**Ingeniería en Sistemas de Computación e Informática**

**Implementación de una guía de buenas prácticas de Integración Continua en proyectos de desarrollo en el área de tecnología de una Empresa Privada.**

Edison P. Mosquera León, Carlos X. Ruiz Águila, David F. Martínez Bravo

**GIT - GITFLOW**

**1.- Instalación de herramientas**

* **Git & Github:**

Ingresamos a la página web de git: <https://git-scm.com/> , descargamos el instalador y lo ejecutamos.

Diagrama

Descripción generada automáticamente

Al momento de instalar dejamos las opciones por defecto que nos ofrece el mismo git.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Procedemos a instalar git.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para Github solo crearemos una cuenta en la misma página web de: <https://github.com/> e ingresamos nuestros datos personales.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**2.- Instalación de dotNet .Net SDK x64:** <https://dotnet.microsoft.com/download>

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

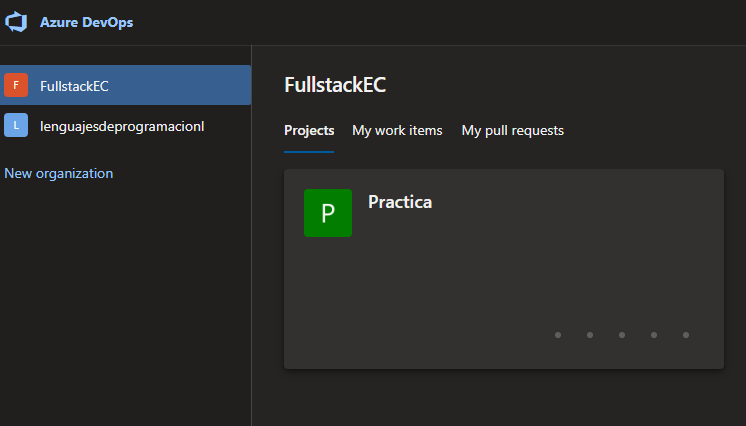
**3.- Flujo CI/CD.**

* Ingresamos al proyecto creado por el Lider del proyecto.

Interfaz de usuario gráfica

Descripción generada automáticamente

* Buscamos el Proyecto, en este caso como es un demo seleccionaremos Practica o ingresaremos mediante: <https://dev.azure.com/FullstackEC/Practica>



* Ingresamos a los repositorios de los microservicios en donde (Numeración está desplegada en la imagen):
  1. Click en “Repos” aquí están todos los repositorios de microservicios.
  2. Click en “Files”, aquí están los archivos de los microservicios.
  3. En esta ventana podemos escoger en cual microservicios queremos enforcarnos.

Captura de pantalla de computadora

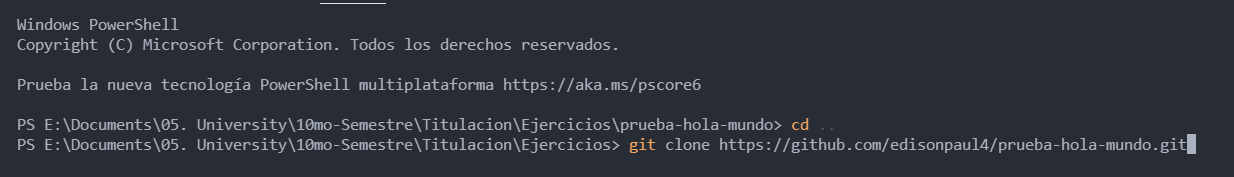
Descripción generada automáticamente

**4.- Clonar el repositorio:**

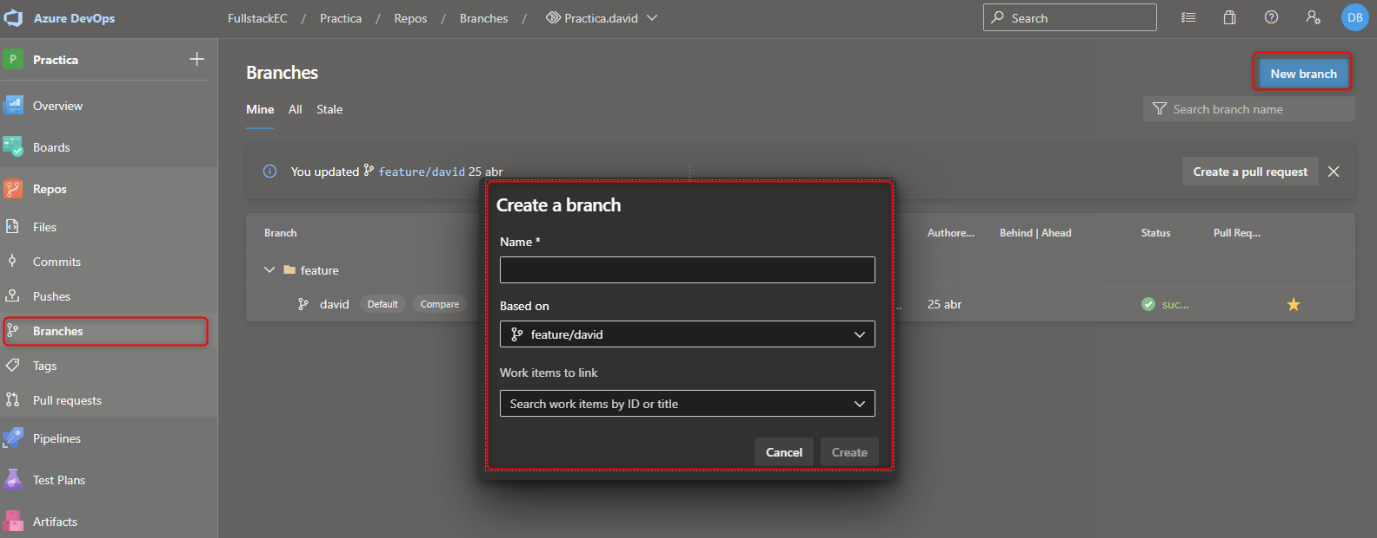
Captura de pantalla de computadora

Descripción generada automáticamente

* Escribimos en nuestra terminal:
  + “git clone <https://github.com/edisonpaul4/prueba-hola-mundo.git>”



* Volvemos a Azure Devops dentro de nuestro repositorio iremos a “Branches” donde crearemos un nuevo branch con el nombre de la tarea que estemos trabajando.



Captura de pantalla de un celular

Descripción generada automáticamente

* Una vez que tengamos listo nuestro feature procederemos a crear un Pull Request:

Captura de pantalla de un celular

Descripción generada automáticamente

* Nos aseguramos de que seleccionemos el branch en el cual están nuestros cambios y posterior el branch hacia donde queremos unir nuestros cambios.

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Captura de pantalla de un celular

Descripción generada automáticamente

* El encargado de revisar los cambios realizados por cada personal del proyecto debe aprobar los Pull Request, aprobarlos y completarlos.
* Una vez que son aprobados pasan de la rama de desarrollo “develop” a la rama principal donde se encuentra la estructura principal del proyecto ya en producción.

Diagrama

Descripción generada automáticamente

* Los primeros pasos es completar la rama “main” con la rama de desarrollo “develop”, mediante los siguientes comandos.

Logotipo

Descripción generada automáticamente con confianza media

* Para esto iniciaremos la biblioteca de extenciones de git-flow en el repositorio, el cual creara la rama de desarrollo:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* Adicionalmente a esto tenemos también otras ramas después de ser codificadas y testeadas que se pueden usar antes de ser enviadas al proyecto principal, son las feature o características.

Diagrama

Descripción generada automáticamente

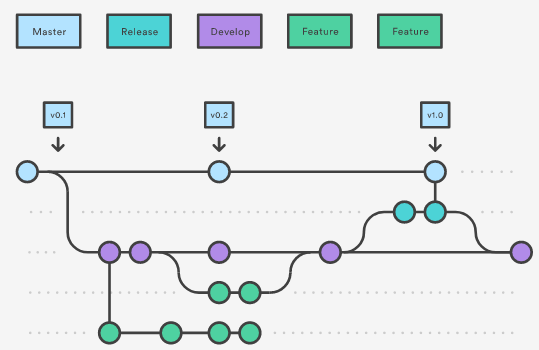
* Estas permiten la fusión con la rama de desarrollo y son las que conforman todos los efectos dentro del flujo de trabajo de la rama función.
* Para crear esta rama dado que estamos con la extensión de git-flow utilizaremos el comando:



* Cuando se aprueban todos los cambios nuevos procederemos a terminar y unir esta rama con la de desarrollo.



* Por último, tenemos la rama de publicación dado que aquí se integran todas las ramas en un único conjunto para todo el proyecto principal.



* Dado a que estamos utilizando el flujo de git-flow nosotros utilizaremos los comandos:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

* Una vez que esté lista permitirá el lanzamiento para que esta rama se fusione con la rama main y la rama de desarrollo dado que esta rama es única para las publicaciones del proyecto principal.

En resumen, esta última parte permitirá un flujo más controlado en el cual están integrados, testeados y aprobados cada una de las diferentes fases para tener un proyecto principal más limpio y mejor estructurado a la hora de sacarlo a producción.