

GIT Aplicado a Github/Bitbucket

Código: GIT-001

Propuesta de Valor: DESARROLLO - PROGRAMACIÓN - METODOLOGÍAS

Duración: 24 Horas



En este curso nos preocupamos que tengas todo para poder cumplir con el aprendizaje de forma completa. Poder desarrollar aplicaciones basadas en el curso Git Management y con la ayuda de una Aplicación BASE completa entregada por el curso podrá entender en formas más reales su aplicación. Con la guía creada para el Alumno será capaz de poder seguir el curso sin problemas.

AUDIENCIA

- Desarrolladores.
- DevOps.

PRE REQUISITOS

- No hay requisitos previos pero es necesario tener conocimientos de desarrollo, programación e informática.

OBJETIVOS

- Adquirir conocimientos sobre GIT.
- Conocer y adquirir habilidades con los comandos básicos en GIT.
- Conocer y adquirir habilidades con los múltiples entornos de trabajo en GIT.

CERTIFICACIÓN DISPONIBLE

- Certificación emitida por COGNOS.



CONTENIDO

1. INTRODUCCIÓN A GIT

- 1.1. ¿QUÉ ES GIT?
- 1.2. INSTALANDO GIT Y GITBASH EN WINDOWS/LINUX
- 1.3. EDITORES DE CÓDIGO, ARCHIVOS BINARIOS Y DE TEXTO PLANO
- 1.4. INTRODUCCIÓN A LA TERMINAL Y LÍNEA DE COMANDOS

2. COMANDOS BÁSICOS EN GIT

- 2.1. CREAR UN REPOSITORIO DE GIT
- 2.2. ANALIZAR CAMBIOS EN LOS ARCHIVOS DE UN PROYECTO CON GIT
- 2.3. ¿QUÉ ES EL STAGING Y LOS REPOSITORIOS? CICLO BÁSICO DE TRABAJO EN GIT
- 2.4. ¿QUÉ ES UN BRANCH (RAMA) Y CÓMO FUNCIONA UN MERGE EN GIT?
- 2.5. VOLVER EN EL TIEMPO EN NUESTRO REPOSITORIO UTILIZANDO RESET Y CHECKOUT
- 2.6. GIT RESET VS. GIT RM

3. FLUJO DE TRABAJO EN GIT

- 3.1. FLUJO DE TRABAJO BÁSICO CON UN REPOSITORIO REMOTO
- 3.2. INTRODUCCIÓN A LAS RAMAS O BRANCHES DE GIT
- 3.3. FUSIÓN DE RAMAS CON GIT MERGE
- 3.4. RESOLUCIÓN DE CONFLICTOS AL HACER UN MERGE

4. TRABAJANDO CON REPOSITORIOS REMOTOS

- 4.1. CAMBIOS EN GITHUB/BITBUCKET: DE MASTER A MAIN
- 4.2. USO DE GITHUB/BITBUCKET
- 4.3. CÓMO FUNCIONAN LAS LLAVES PÚBLICAS Y PRIVADAS
- 4.4. CONFIGURA TUS LLAVES SSH EN LOCAL
- 4.5. CONEXIÓN A GITHUB CON SSH
- 4.6. TAGS Y VERSIONES EN GIT, GITHUB Y BITBUCKET
- 4.7. MANEJO DE RAMAS EN GITHUB/BITBUCKET
- 4.8. CONFIGURAR MÚLTIPLES COLABORADORES EN UN REPOSITORIO DE GITHUB/BITBUCKET

5. FLUJOS DE TRABAJO

- 5.1. HACIENDO MERGE DE RAMAS DE DESARROLLO A MASTER
- 5.2. FLUJO DE TRABAJO CON PULL REQUESTS
- 5.3. UTILIZANDO PULL REQUESTS EN GITHUB/BITBUCKET
- 5.4. CREANDO UN FORK, CONTRIBUYENDO A UN REPOSITORIO
- 5.5. HACIENDO DEPLOYMENT A UN SERVIDOR
- 5.6. HAZME UN PULL REQUEST
- 5.7. IGNORAR ARCHIVOS EN EL REPOSITORIO CON .GITIGNORE
- 5.8. README.MD ES UNA EXCELENTE PRÁCTICA

6. MÚLTIPLES ENTORNOS DE TRABAJO EN GIT

- 6.1. GIT REBASE: REORGANIZANDO EL TRABAJO REALIZADO
- 6.2. GIT STASH: GUARDAR CAMBIOS EN MEMORIA Y RECUPERARLOS DESPUÉS
- 6.3. GIT CLEAN: LIMPIAR TU PROYECTO DE ARCHIVOS NO DESEADOS
- 6.4. GIT CHERRY-PICK: TRAER COMMITS VIEJOS AL HEAD DE UN BRANCH

7. COMANDOS GIT EN CASO DE PROBLEMAS

- 7.1. RECONSTRUIR COMMITS EN GIT CON AMEND
- 7.2. GIT RESET Y REFLOG: ÚSESE EN CASO DE EMERGENCIA
- 7.3. BUSCAR EN ARCHIVOS Y COMMITS DE GIT CON GREP Y LOG

★ BENEFICIOS

- Al finalizar el curso, tendrás conocimientos y habilidades en GIT.