Ingeniería de Software Colaborativa

Git, GitHub/GitLab y CI/CD para el Mundo Real

Clase 0 · Bienvenida al Curso

GLUD — Grupo GNU/Linux
Universidad Distrital

Curso Complementario · 10 Semanas

Mapa de ruta

1. Presentación
2. Sobre el Curso
3. Cronograma
4. Metodología
5. Recursos y Apoyo
6. Preparación para la Próxima Clase

1

Presentación

¿Quién soy?

Tu Instructor

Estudiante de Ingeniería de Sistemas

Miembro activo de GLUD

Grupo GNU/Linux Universidad Distrital

¿Por qué estoy aquí?

- Experiencia práctica con Git/GitHub
- Certificación profesional en Git
- Proyectos colaborativos reales

¿Quién soy?

Tu Instructor

Estudiante de Ingeniería de Sistemas

Miembro activo de GLUD

Grupo GNU/Linux Universidad Distrital

¿Por qué estoy aquí?

- Experiencia práctica con Git/GitHub
- Certificación profesional en Git
- Proyectos colaborativos reales

Certificación

Platzi - Git & GitHub

Formación profesional validada

¿Por qué este curso?

La Realidad

Lo que NO te enseñan en clase:

- Trabajo colaborativo asíncrono
- Control de versiones profesional
- Flujos de trabajo en equipos
- Automatización de procesos
- Code Review y mejores prácticas

¿Por qué este curso?

La Realidad

Lo que NO te enseñan en clase:

- Trabajo colaborativo asíncrono
- Control de versiones profesional
- Flujos de trabajo en equipos
- Automatización de procesos
- Code Review y mejores prácticas

Lo que aprenderás aquí

Herramientas del mundo real:

- Git desde cero hasta avanzado
- GitHub/GitLab como profesional
- Pipelines de CI/CD
- Trabajo en equipo simulado
- Proyectos con evidencia auditable

¿Por qué este curso?

La Realidad

Lo que NO te enseñan en clase:

- Trabajo colaborativo asíncrono
- Control de versiones profesional
- Flujos de trabajo en equipos
- Automatización de procesos
- Code Review y mejores prácticas

Lo que aprenderás aquí

Herramientas del mundo real:

- Git desde cero hasta avanzado
- GitHub/GitLab como profesional
- Pipelines de CI/CD
- Trabajo en equipo simulado
- Proyectos con evidencia auditable

Este curso cierra la brecha entre la academia y la industria

> *La mejor inversión que puedes hacer como desarrollador es aprender a trabajar en equipo usando las herramientas correctas.*
>
> — Industria del Software

2

Sobre el Curso

Información del Curso

ISC-GIT-101

Ingeniería de Software Colaborativa: Git y CI/CD

Duración y Formato:

- **10 semanas** (≈40 horas)
- Sesiones teórico-prácticas
- Proyecto colaborativo real

Tipo de Curso:

- Complementario
- Electivo

Información del Curso

ISC-GIT-101

Ingeniería de Software Colaborativa: Git y CI/CD

Duración y Formato:

- **10 semanas** (≈40 horas)
- Sesiones teórico-prácticas
- Proyecto colaborativo real

Tipo de Curso:

- Complementario
- Electivo

Requisitos:

- Conocimientos básicos de programación
- Cualquier lenguaje (Java, Python, JS...)
- Computador/VM con Linux

Certificación:

- Validación por evidencia
- Repositorio auditado
- 70/100 para aprobar

Objetivo General

¿Qué vas a lograr?

Al finalizar este curso serás capaz de:

- **Gestionar proyectos profesionalmente** usando Git y GitHub/GitLab
- **Trabajar en equipos** siguiendo flujos colaborativos estándar de la industria
- **Automatizar procesos** de construcción y pruebas con pipelines CI/CD
- **Aplicar mejores prácticas** de ingeniería de software en cualquier stack tecnológico
- **Demostrar competencias** con evidencia auditable de trabajo colaborativo

¿Qué aprenderás exactamente?

Competencias Técnicas:

- Configurar entornos profesionales
- Dominar Git (local y remoto)
- Gestión de ramas y conflictos
- Pipelines de CI/CD automáticos
- Integración con herramientas (Maven, npm, pytest...)

Estándares de Industria:

- Conventional Commits
- Semantic Versioning
- Git Flow
- Code Review

¿Qué aprenderás exactamente?

Competencias Técnicas:

- Configurar entornos profesionales
- Dominar Git (local y remoto)
- Gestión de ramas y conflictos
- Pipelines de CI/CD automáticos
- Integración con herramientas (Maven, npm, pytest...)

Estándares de Industria:

- Conventional Commits
- Semantic Versioning
- Git Flow
- Code Review

Competencias Colaborativas:

- Trabajo en equipos multidisciplinarios
- Roles rotativos (Tech Lead/Developer)
- Comunicación técnica efectiva
- Pull Requests y revisiones
- Gestión de Issues y Milestones

Preparación Profesional:

- Flujos DevOps
- Metodologías ágiles
- Entrega continua
- Portfolio demostrable

3

Cronograma

Módulo 1: Fundamentos (Semanas 1-3)

1 Semana 1

Configuración

- Git + herramientas
- Variables de entorno
- Arquitectura de Git
- Primer `git init`

2 Semana 2

Ciclo de Vida

- Estados del código
- Conventional Commits
- Estructura de proyecto
- `.gitignore`

3 Semana 3

Historial

- `git log` avanzado
- `git diff`
- `git restore`
- Tags y versiones

Módulo 1: Fundamentos (Semanas 1-3)

1 Semana 1
Configuración

- **Git + herramientas**
- **Variables de entorno**
- **Arquitectura de Git**
- **Primer** `git init`

2 Semana 2
Ciclo de Vida

- **Estados del código**
- **Conventional Commits**
- **Estructura de proyecto**
- `.gitignore`

3 Semana 3
Historial

- `git log` **avanzado**
- `git diff`
- `git restore`
- **Tags y versiones**

Entrega Parcial 1

Repositorio inicial con estructura de proyecto y primeros commits semánticos

Módulo 2: Colaboración (Semanas 4-6)

4 Semana 4

Branching

- Concepto de ramas
- Punteros y referencias
- `feature/*` branches
- Por qué no usar `main`

5 Semana 5

Conflictos

- Merge vs Rebase
- Resolución de conflictos
- Laboratorio de Caos
- Estrategias de fusión

6 Semana 6

Trabajo Remoto

- Autenticación SSH
- GitHub/GitLab
- Clone, Push, Pull
- Git Flow completo

Módulo 2: Colaboración (Semanas 4-6)

4 Semana 4

Branching

- Concepto de ramas
- Punteros y referencias
- `feature/*` branches
- Por qué no usar `main`

5 Semana 5

Conflictos

- Merge vs Rebase
- Resolución de conflictos
- Laboratorio de Caos
- Estrategias de fusión

6 Semana 6

Trabajo Remoto

- Autenticación SSH
- GitHub/GitLab
- Clone, Push, Pull
- Git Flow completo

Hito Importante

Equipos formados, roles asignados, proyecto colaborativo en marcha

Módulo 3: Calidad (Semanas 7-8)

7

Semana 7

Code Review

- Pull Requests (PR)
- Revisión de código
- Aprobaciones
- Rol de Tech Lead

8

Semana 8

Project Management

- Issues y Labels
- Milestones
- Tableros Kanban
- Vinculación con commits

Módulo 3: Calidad (Semanas 7-8)

7

Semana 7

Code Review

- Pull Requests (PR)
- Revisión de código
- Aprobaciones
- Rol de Tech Lead

8

Semana 8

Project Management

- Issues y Labels
- Milestones
- Tableros Kanban
- Vinculación con commits

Entrega Parcial 2

Demo intermedia: proyecto con PRs documentados y gestión de Issues

Módulo 4: Automatización (Semanas 9-10)

9

Semana 9

CI - Integración Continua

- ¿Qué es un Pipeline?
- Estructura YAML
- GitHub Actions / GitLab CI
- Automatizar tests
- Bloqueo de PRs fallidos

10

Semana 10

CD - Despliegue Continuo

- Generación de artifacts
- Publicación en Releases
- Tags y versiones
- **Proyecto Final**
- Auditoría GLUD

Módulo 4: Automatización (Semanas 9-10)

9

Semana 9

CI - Integración Continua

- ¿Qué es un Pipeline?
- Estructura YAML
- GitHub Actions / GitLab CI
- Automatizar tests
- Bloqueo de PRs fallidos

10

Semana 10

CD - Despliegue Continuo

- Generación de artifacts
- Publicación en Releases
- Tags y versiones
- **Proyecto Final**
- Auditoría GLUD

Entrega Final

Repositorio auditado con pipeline CI/CD funcional

4

Metodología

¿Cómo funcionará el curso?

Simulación de Entorno Laboral

Este NO es un curso tradicional. Es una experiencia práctica simulando una empresa real.

Equipos de 3 personas:

- Roles rotativos semanales
- **Tech Lead:** Revisa y aprueba código
- **Developers:** Implementan features

Cada sesión:

- 30-50 min teoría
- 60-70 min práctica
- Trabajo colaborativo en tiempo real

¿Cómo funcionará el curso?

Simulación de Entorno Laboral

Este NO es un curso tradicional. Es una experiencia práctica simulando una empresa real.

Equipos de 3 personas:

- Roles rotativos semanales
- **Tech Lead:** Revisa y aprueba código
- **Developers:** Implementan features

Cada sesión:

- 30-50 min teoría
- 60-70 min práctica
- Trabajo colaborativo en tiempo real

Proyecto colaborativo:

- Stack tecnológico a elección
- Desarrollo incremental
- Entregas parciales
- Pipeline CI/CD funcional

Apoyo del GLUD:

- Plantillas base
- Auditoría final

Stack Tecnológico Flexible

Agnóstico del Lenguaje

Puedes usar el lenguaje que prefieras. El curso se adapta a tu stack.

Sugerencia inicial:

Java + Maven

Java JDK 17+
Apache Maven

- **Fácil integración CI/CD**
- **Comandos estándar (`mvn test`)**
- **Ecosistema maduro**

Stack Tecnológico Flexible

Agnóstico del Lenguaje

Puedes usar el lenguaje que prefieras. El curso se adapta a tu stack.

Sugerencia inicial:

Java + Maven

Java JDK 17+
Apache Maven

- **Fácil integración CI/CD**
- **Comandos estándar (`mvn test`)**
- **Ecosistema maduro**

Otras opciones válidas:

- **Python** + pytest/unittest
- **JavaScript/TypeScript** + npm/Jest
- **C++** + CMake/Google Test
- **C#** + .NET
- Cualquier lenguaje con tests automatizables

Lo importante es que tenga build tool y testing framework

5

Recursos y Apoyo

¿Qué recursos tendrás?

Material del GLUD:

- Slides de cada clase
- Plantillas de proyectos
- Guías de configuración
- Templates de CI/CD
- Ejemplos de buenas prácticas
- Cheatsheets de comandos

Plataformas:

- GitHub/GitLab (hosting)
- Git Bash/Terminal
- VS Code (recomendado)

¿Qué recursos tendrás?

Material del GLUD:

- Slides de cada clase
- Plantillas de proyectos
- Guías de configuración
- Templates de CI/CD
- Ejemplos de buenas prácticas
- Cheatsheets de comandos

Plataformas:

- GitHub/GitLab (hosting)
- Git Bash/Terminal
- VS Code (recomendado)

Documentación oficial:

- `git-scm.com/doc`
- GitHub Learning Lab
- GitLab Documentation
- Atlassian Git Tutorials

Práctica interactiva:

- `learngitbranching.js.org`
- Git Immersion
- GitHub Skills

Apoyo continuo:

- Comunidad del curso
- Sesiones de Q&A

Formación de Equipos

Próximo Paso

Al final de esta sesión formaremos los equipos de 3 personas.

1. Considera tus fortalezas y debilidades
2. Busca complementariedad en tu equipo
3. Acuerden stack tecnológico (sugerencia: Java + Maven)
4. Definan canales de comunicación

Formación de Equipos

Próximo Paso

Al final de esta sesión formaremos los equipos de 3 personas.

1. Considera tus fortalezas y debilidades
2. Busca complementariedad en tu equipo
3. Acuerden stack tecnológico (sugerencia: Java + Maven)
4. Definan canales de comunicación

El éxito del proyecto depende del **trabajo en equipo**, no solo de habilidades individuales

6

Preparación para la Próxima Clase

Recomendaciones Antes de la Clase 1

Tareas antes de Clase 1

Para aprovechar al máximo la siguiente sesión:

- ☐ Formar equipo de 3 personas
- ☐ Decidir stack tecnológico (Java + Maven recomendado)
- ☐ Crear cuenta en GitHub o GitLab
- ☐ (Opcional y recomendado) Traer computador con Linux instalado
- ☐ (Opcional) Explorar `learngitbranching.js.org`
- ☐ Leer el Syllabus completo

Clase 1

Instalación de Git + herramientas del stack. Variables de entorno. Arquitectura de Git. Primer `git init`.

¡Trae tu computador listo para trabajar!