

Programa de Certificado MasterTrack™

Principios de ingeniería de software automatizada y ágil

Ciclo 1

Curso 1:

Prácticas de ingeniería de software esenciales para el agilismo

Número de horas a la semana:

2 meses: curso 1 y 2 en simultáneo (16-24 horas por semana de estudio)

Descripción:

Utiliza prácticas esenciales de ingeniería de software para ser un miembro efectivo y eficiente en un equipo de desarrollo ágil (la documentación de requisitos en historias de usuario, flujos de trabajo sobre un manejador de versiones, el desarrollo orientado a pruebas y la práctica de la integración continua).

Tecnologías principales:

- El lenguaje de programación es Python
- Para el versionamiento: Git y Github
- Para la integración continua: Jenkins

Lo que aprenderás

- Escribe historias de usuario que cumplan las características INVEST. (Independientes, negociable, valiosas, estimables, pequeñas y que se puedan probar) a partir de un enunciado para el desarrollo de un producto de software de tamaño pequeño.
- Utiliza efectivamente un manejador de versiones y seguir un flujo de trabajo definido para sincronizar su desarrollo con el de un equipo.
- Realiza TDD (Test Driven Development) para escribir pruebas unitarias y código.
- Utiliza las prácticas de la Integración continua como la construcción automática de los ejecutables, el manejo de dependencia, la ejecución automática de pruebas y los tableros de control para hacer seguimiento.
- Configura una herramienta de IC para automatizar los procesos de construcción, ejecución automática de pruebas y despliegue en un proyecto de desarrollo de un producto de software de tamaño pequeño.
- Analiza los resultados del proceso de IC para ajustar el proyecto.

Metodología del curso:

El curso combina teoría y práctica. La teoría se ofrece a través de vídeos y lecturas y se complementa con testimonios de industria, con personas que aportarán desde su experiencia.

La práctica se desarrolla a través de tutoriales, talleres individuales en cada tema y se complementa con un proyecto. Es un proyecto pequeño para realizar en parejas donde el estudiante tendrá que definir historias de usuario que cumplan las características INVEST (Independientes, negociable, valiosas, estimables, pequeñas y que se puedan probar), con base en las historias de usuario, utiliza TDD (Test Driven Development) para construir las pruebas unitarias y el código. Esta construcción sigue un proceso de integración continua apoyado en herramientas.