

Git (comandos):

Git es un sistema de control de versiones que permite a los desarrolladores colaborar en proyectos de software de manera efectiva. Algunos comandos importantes que debes conocer incluyen:

- git clone: clona un repositorio desde una URL remota.
- git add: agrega cambios a la zona de preparación.
- git commit: guarda cambios en el repositorio.
- git push: envía cambios al repositorio remoto.
- git pull: obtiene cambios del repositorio remoto y los fusiona con tu copia local.

Es importante conocer estos comandos básicos, así como otros comandos útiles que pueden ayudarte a trabajar con Git de manera más eficiente.

GitHub:

GitHub es una plataforma web que utiliza Git como su sistema de control de versiones. Permite a los desarrolladores colaborar en proyectos de software, compartir código y rastrear problemas. Algunas funciones útiles que debes conocer incluyen:

- Repositorios: donde se almacena el código fuente del proyecto.
- Issues: donde se rastrean los problemas y errores.
- Pull requests: donde los desarrolladores proponen cambios al proyecto.
- Colaboradores: personas que tienen acceso al proyecto y pueden contribuir al código.

Es importante comprender cómo usar estas funciones para colaborar efectivamente en proyectos en GitHub.

GitHub Actions:

GitHub Actions es un servicio de integración continua que permite a los desarrolladores automatizar tareas comunes en el desarrollo de software. Algunas funciones útiles que debes conocer incluyen:

Crear flujos de trabajo para automatizar tareas como la construcción y pruebas.

Utilizar acciones predefinidas o crear tus propias acciones personalizadas para realizar tareas específicas.

Configurar desencadenantes para ejecutar flujos de trabajo automáticamente, como cuando se realiza un push a una rama determinada.

Es importante comprender cómo configurar flujos de trabajo y utilizar las acciones de GitHub para automatizar tareas comunes en el desarrollo de software.

SemVer:

SemVer (Versión Semántica) es un sistema de numeración de versiones que se utiliza para el software. Se compone de tres números, por ejemplo, 1.2.3, que indican la versión principal, la versión secundaria y la versión de corrección. Es importante comprender cómo utilizar SemVer para versionar tu software correctamente.

Manifiesto Ágil (valores, principios)

El Manifiesto Ágil es un conjunto de valores y principios que guían la gestión de proyectos de software ágil. Los cuatro valores del Manifiesto Ágil son:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación exhaustiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta al cambio sobre seguir un plan.

Es importante comprender estos valores y principios y cómo se aplican en la gestión de proyectos de software ágil.

Prácticas Ágiles:

Las prácticas ágiles son un conjunto de técnicas y herramientas que se utilizan para gestionar proyectos de software ágil. Algunas prácticas útiles que debes conocer incluyen:

- Planificación iterativa y incremental.
- Entrega continua de software
- Integración continua (CI).
- Pruebas automatizadas.
- Refactorización constante.
- Diseño simple.
- Trabajo en equipo colaborativo.
- Retrospectivas regulares.

Es importante comprender estas prácticas y cómo se aplican en la gestión de proyectos de software ágil.

DevOps:

DevOps es una metodología que combina la gestión de desarrollo de software (Dev) y la gestión de operaciones de infraestructura (Ops) en un proceso de desarrollo continuo. Algunos conceptos importantes que debes conocer incluyen:

- Automatización de procesos de desarrollo y operaciones.
- Integración continua y entrega continua (CI/CD).
- Monitoreo y análisis de rendimiento en tiempo real.
- Implementación segura y confiable de cambios.

Es importante comprender cómo se aplican estos conceptos en la metodología DevOps y cómo pueden mejorar la eficiencia y la calidad del proceso de desarrollo de software.

Kanban:

Kanban es un marco de trabajo ágil que se utiliza para gestionar el flujo de trabajo en un proyecto de software. Algunos conceptos importantes que debes conocer incluyen:

- Tableros Kanban que representan el flujo de trabajo del proyecto.
- Tarjetas Kanban que representan las tareas y los elementos del trabajo.
- Limitar el trabajo en progreso (WIP).
- Priorizar las tareas por valor para el cliente.
- Medir y mejorar el flujo de trabajo.

Es importante comprender cómo se aplica Kanban en la gestión de proyectos de software y cómo puede mejorar la eficiencia y la calidad del proceso.

Scrum (artefactos, eventos, roles, pilares):

Scrum es un marco de trabajo ágil que se utiliza para gestionar proyectos de software. Algunos conceptos importantes que debes conocer incluyen:

- Artefactos: Product Backlog, Sprint Backlog y Incremento.
- Eventos: Sprint, Reunión de planificación del Sprint, Reunión diaria, Revisión del Sprint y Retrospectiva del Sprint.
- Roles: Scrum Master, Dueño del Producto y Equipo de Desarrollo.
- Pilares: Transparencia, Inspección y Adaptación.

Es importante comprender cómo se aplican estos conceptos en Scrum y cómo pueden mejorar la eficiencia y la calidad del proceso de desarrollo de software.

ORM:

ORM (Mapeo Objeto-Relacional) es una técnica de programación que se utiliza para mapear objetos de un lenguaje de programación a tablas de una base de datos relacional. Algunos conceptos importantes que debes conocer incluyen:

ORM frameworks como Hibernate o Entity Framework.

Mapeo de objetos a tablas de base de datos.

Uso de lenguaje de consulta (por ejemplo, SQL) para realizar operaciones en la base de datos.

Es importante comprender cómo se aplica ORM en el desarrollo de software y cómo puede mejorar la eficiencia y la calidad del proceso de desarrollo.

Azure Portal:

Azure Portal es un portal web que se utiliza para administrar recursos y servicios en la nube de Microsoft Azure. Algunos conceptos importantes que debes conocer incluyen:

- Creación y gestión de máquinas virtuales.
- Configuración de redes y seguridad.
- Configuración y monitoreo de aplicaciones web.
- Configuración y gestión de servicios de bases de datos.

Es importante comprender cómo se utiliza Azure Portal para administrar los recursos y servicios en la nube de Microsoft Azure y cómo se integra en el proceso de desarrollo de software.

Azure DevOps:

Azure DevOps es un conjunto de servicios y herramientas que se utilizan para desarrollar, probar y entregar software en la nube de Microsoft Azure. Algunos conceptos importantes que debes conocer incluyen:

- Control de versiones de código fuente con Git.
- Integración continua y entrega continua (CI/CD).
- Gestión de proyectos y seguimiento de trabajo con tableros Kanban y Scrum.
- Pruebas automatizadas y monitoreo de rendimiento.
- Implementación segura y confiable de cambios.

Es importante comprender cómo se utiliza Azure DevOps para mejorar el proceso de desarrollo de software y cómo se integra con otras prácticas y metodologías ágiles.