



Práctica #1.

Parte I: Introducción a Git y Visual Code

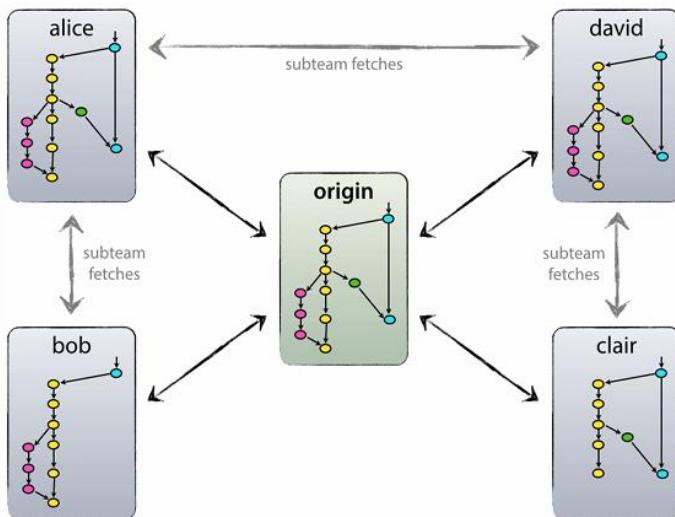
Git

Es una herramienta de control de versiones, sirve para gestionar cambios en el código fuente de los programas y poder revertirlos. Git permite:

- Colaboración entre desarrolladores (reúso de componentes)
- Almacenar versiones (correctamente)
- Recuperar versiones previas
- Entender qué pasó (quién modificó, cuándo y por qué)
- Respaldo

Conceptos importantes:

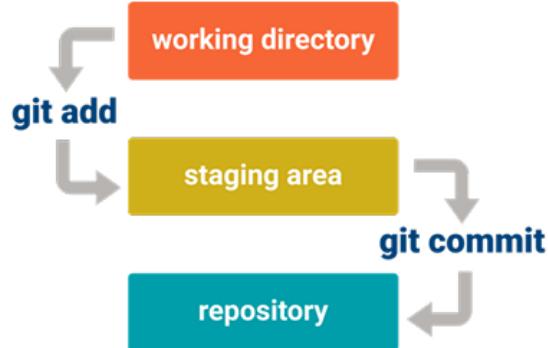
- Sistema Distribuido:



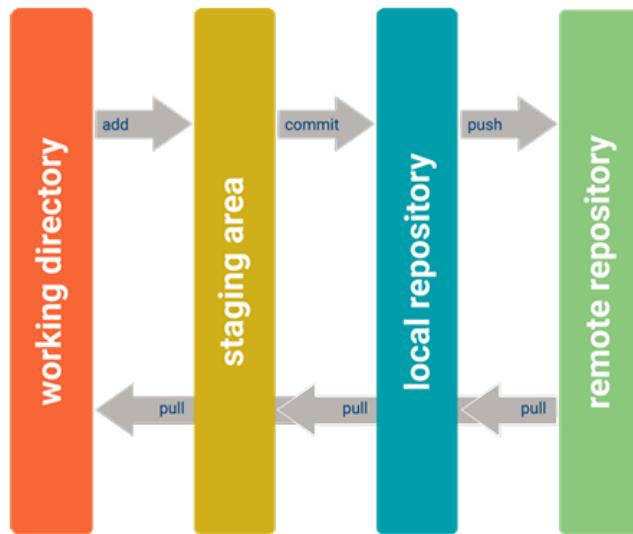
- Distintos estados de un archivo:



- Diferencia entre el directorio de trabajo, área intermedia y repositorio:



- Agregar, comprometer, subir y bajar archivos de un repositorio remoto:



- Administración de ramas: permite rapidez en la gestión de ramas (branch) y la unificación de diferentes versiones (merge)



GitHub

Es una plataforma de desarrollo colaborativo que permite hospedar repositorios del sistema de control de versiones Git, gestionar proyectos y hacer revisiones de código. Se utiliza principalmente para la creación de código fuente de programas de computadora. Gratuito para uso

público, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

URL: <https://github.com/>

GitHub ofrece por proyecto una wiki, una página web, un gráfico para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto. También tiene funcionalidades como si se tratase de una red social, por ejemplo, seguidores. Es una herramienta para trabajo colaborativo entre programadores y posee un gestor de proyectos de estilo Kanban.

Ejercicio:

1. Crear una cuenta en GitHub y un proyecto llamado IS-PRACTICAS
2. Instalar Visual Code
3. Instalar extensión Git Graph
4. Desde Visual Code clonar el proyecto creado en el paso 1
5. Crear la estructura de directorios a usar:
 - docs/ -> directorio la documentación
 - docs/diagrams -> directorio para la fuente de los diagramas UML
 - src -> directorio para guardar el código de la aplicación
 - src/main -> directorio para el código fuente
 - src/test -> directorio para las pruebas unitarias
 - README.md -> archivo readme del proyecto
6. Editar el archivo README.md o modificarlo
7. Crear archivo .gitignore y agregar archivos que no requieran seguimiento
8. Guardar los cambios con un commit y un mensaje “estructura inicial”
9. Subir los cambios al repositorio remoto (publicar rama)
10. Revisar en GitHub que los cambios estén publicados
11. Modificar el archivo README.md desde GitHub
12. Actualizar código local con el cambio hecho en el GitHub (sincronizar)
13. Crear una rama desde la rama main llamada “probando rama”
14. Crear el archivo “diagrama.plantuml”, en el directorio “docs/diagrams”
15. Hacer commit con el mensaje “diagrama inicial”
16. Subir rama a GitHub y revisar que este ahí
17. Cambiar a la rama main y hacer un merge con la rama “probando rama”
18. Subir el merge a GitHub
19. Regresar desde VisualCode a la rama “probando rama” y cambiar una línea en el archivo
20. Hacer commit del cambio de la rama con mensaje “cambio a diagrama” y subir al repositorio de GitHub
21. Borrar la rama “probando rama”
22. Subir todos los cambios a GitHub
23. Revisar en GitHub todo
24. Ver historial de Git desde Visual Code usando la extensión Git Graph

Parte II: Unified Modeling Language

UML

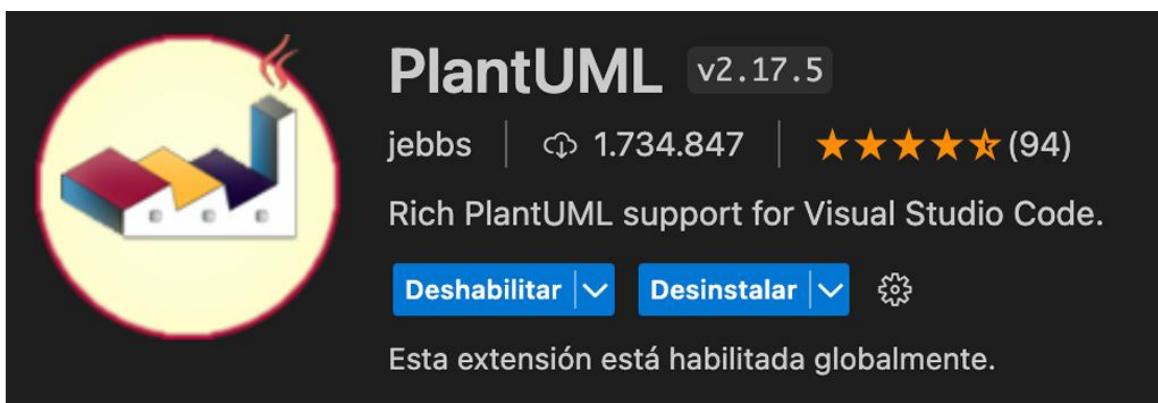
Proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas. Hay cuatro tipos de elementos: estructurales, de comportamiento, de agrupación y de anotación. Hay 7 tipos principales de elementos estructurales: casos de usos, clases, clases activas (clase cuyas instancias son objetos activos), interfaces, componentes, colaboraciones y nodos. Hay 2 tipos de elementos de comportamiento: interacciones y máquinas de estado. Hay 4 tipos de agrupaciones: paquetes, modelos, subsistemas y marcos de trabajo. Solo hay un tipo de elemento de anotación: notas.

PlantUML

es un generador de diagramas UML a partir de la descripción textual de cada elemento del diagrama y sus relaciones. Es una aplicación web que te permite a partir de unas instrucciones en un archivo de texto generar distintos diagramas UML

URL: <https://plantuml.com/es/>

Desde Visual Code podemos usar esta herramienta al instalar la extensión PlantUML. Esta extensión da soporte para editar los archivos de extensión “.plantuml”, ver previsiones de los diagramas y exportarlos en formato de imagen.



Ejercicios: Modele las siguientes descripciones y conceptos utilizando un diagrama de clase:

1. Un teclado tiene por lo menos 105 teclas
2. Los estudiantes, los profesores y los empleados son personas
3. Un disco duro está formado por pistas, y las pistas están formadas por sectores
4. Un triángulo es un polígono que está formado por 3 lados
5. Un equipo de baseball está formado por jugadores y tienen asignado como homeclub un estadio
6. Un comité está integrado por dos o más personas. Una persona puede o no pertenecer a comités. Cada comité tiene un coordinador que forma parte del comité y una misma persona puede coordinar más de un comité.

8. Una caracterización se fundamenta en al menos un criterio, un elemento clasificado se genera a partir de una caracterización que depende del criterio utilizado.
9. Un trabajo de grado especial (TEG) y un trabajo de grado de maestría (TGM) son trabajos de investigación. Todo trabajo de investigación es dirigido por uno o a lo sumo dos profesores que juegan el rol de tutor. Un mismo profesor puede dirigir varios trabajos de investigación. Un TGM siempre se realiza de manera individual, es decir, un solo estudiante de postgrado mientras que un TEG puede ser realizado por uno o dos estudiantes de pregrado.