

Metodología de trabajo con Git - DESIN

Concepto	Descripción
GITHUB	Espacio para almacenamiento de repositorios en la nube
GITHUB. Organización	Cada módulo tendrá una organización asociada
GITHUB. Organización. Miembros	Todo el alumnado del módulo pertenece a la organización
GITHUB. Organización. Miembros. Repositorio	Cada estudiante creará un repositorio con el formato MÓDULO-CURSO_CICLO-2024-2025-Nombre_Apellido

RAMAS	
RAMA main	Rama para las entregas finales de actividades o tareas
RAMA stage	Rama para entregas parciales para su revisión antes de la entrega final (cuando proceda)
RAMA develop	Rama para fusionar funcionalidades entre miembros del equipo (trabajos en grupo)
Cada tarea / ejercicio	
RAMA feature/udx_acty	<p>Por cada actividad se creará una rama para trabajar. Esta rama se mergeará en stage para hacer pre-entregas o main para hacer la entrega final</p> <p><i>Ejemplo:</i> Rama feature/ud1_act2</p>
Carpeta udx/actividades/udx_acty	Dentro de cada rama de feature, los estudiantes crearán una carpeta donde se guardarán todos los archivos asociados a la tarea o ejercicio, siguiendo la misma convención empleada hasta ahora.
Cada unidad (ejemplos)	
RAMA feature/ejemplos_udx	Por cada unidad, se creará una rama para los ejemplos de dicha actividad. Dicha rama se mergeará en stage a medida que vayamos avanzando y en main al acabar la unidad.
Carpeta udx/ejemplos/y_ejemplo	Dentro de cada rama de feature, los estudiantes crearán una carpeta con el mismo identificador donde se guardarán todos los archivos asociados a la tarea o ejercicio, siguiendo la misma convención empleada hasta ahora.

FLUJO DE TRABAJO. Nueva tarea / ejercicio	
1. Creación de rama para su desarrollo	
RAMA feature/udx_acty	
Desde la rama main	<code>git checkout -b feature/udx_acty</code>
2. Creación de carpeta asociada a la tarea/ejercicio	
Carpeta udx/ejercicios/udx_acty	
Todo el desarrollo de la tarea o ejercicio se guardará en la carpeta asociada.	
3. Trabajo en la tarea/ejercicio	
Todo el trabajo de la tarea/ejercicio se hace en la rama correspondiente. Cada día de trabajo se deberá hacer un commit con lo trabajado	
4. Pre-entrega (si procede)	
En el caso de que se requiera una revisión por parte del profesorado de la tarea o ejercicio	
Merge de la rama feature/udx_acty a stage	
Desde la rama stage	<code>git merge feature/udx_acty</code>
5. Entrega	
Una vez finalizada la tarea/ejercicio se hace la entrega	
Merge de la rama feature/udx_acty a main	
Desde la rama main	<code>git merge feature/udx_acty</code>

Comandos más habituales de GIT	
<i>git add nombre_archivo</i>	Añadir un archivo específico al repositorio
<i>git add .</i>	Añadir todos los archivos al repositorio
<i>git commit -m "DESCRIPCIÓN"</i>	Guardar una versión del código con un comentario descriptivo
<i>git push origin RAMA</i>	Subir la rama actual al repositorio remoto
<i>git pull origin RAMA</i>	Descargar la última versión de la rama remota
<i>git checkout RAMA</i>	Cambiarse a una rama específica
<i>git checkout -b RAMA</i>	Crear una rama a partir de la actual
<i>git status</i>	Ver el estado actual del repositorio (archivos modificados, pendientes de añadir, etc.).
<i>git log</i>	Ver el historial de commits del repositorio
<i>git branch</i>	Listar todas las ramas del repositorio
<i>git branch -d RAMA</i>	Eliminar una rama que ya no se necesita (después de hacer el merge)
<i>git reset --soft HEAD~1</i>	Deshacer el último commit pero mantener los cambios en el área de trabajo

<i>git reset -hard HEAD~1</i>	Deshacer el último commit y eliminar los cambios del área de trabajo. (Cuidado: irreversible)
<i>git stash</i>	Guardar temporalmente los cambios sin hacer commit, útil cuando se quiere cambiar de rama pero no se desea hacer commit
<i>git stash apply</i>	Recuperar los cambios guardados con git stash