

ACOGIDA – explicar (no puntuable)

→ ENTORNOS DE DESARROLLO

- Organización inicial en Git:

1. Cada grupo creará un repositorio remoto EN FORMATO ABIERTO-PÚBLICO (GitHub).
2. Configuración adecuada del repositorio, archivos ignore, attributes....
3. Configurar acceso a todos los miembros (colaboradores).
4. **Establecer la creación de ramas, cada rama un usuario. Tener en cuenta la principal**

→ Se debe enseñar en el documento de creación UN LOG del proceso de commits de los usuarios, las ramas utilizadas y explicar que ha realizado cada alumno

→ Explicar cómo se ha solucionado la resolución de conflictos entre ramas.

Como van a separar las tareas para subir los datos en git, el trabajo debe ser en grupo y algunas partes se deben modificar a futuro. Se debe subir los estados de creación de los diagramas, es decir, como ha avanzado el proyecto a lo largo del tiempo

- **DIAGRAMAS DE CLASES y COMPORTAMIENTO:**

1. Crear los diagramas de comportamiento, siendo los de **CASOS DE USO, SECUENCIA Y COLABORACIÓN**
2. Se deben crear los diagramas de clases correspondientes, una vez realizados los de comportamiento. **EL DIAGRAMA DE CLASES CONTIENE LAS CLASES DE LA BASE DE DATOS Y DE LA LÓGICA DEL PROGRAMA**
3. Deben ser acordes al programa a realizar en programación y bases de datos, lo suficientemente largos para tener funcionalidades.

IDEAR Presentación de idea y organización de tareas

→ ENTORNOS DE DESARROLLO

- **DIAGRAMAS DE COMPORTAMIENTO INICIALES:**

→ Crear diagramas de casos de uso básicos.

- ¿Qué acciones realizará un usuario? (Ej: "Registrar usuario", "Activar servicio", "Consultar estado del servidor").
- ¿Qué casos de uso principales existen?

→ Crear diagramas de SECUENCIA Y COLABORACIÓN, una vez realizado el de casos de uso.

Tener una visión general de los principales objetos y relaciones que se necesitarán (aunque luego cambie).:

- Definir clases básicas.
- Establecer atributos generales
- Dibujar relaciones simples (asociaciones, agregaciones).

EXPLORAR Selección de tecnologías y herramientas

→ ENTORNOS DE DESARROLLO

→ **Refinamiento de Comportamiento DETALLADO**

Diagrama de Comportamiento: Crear diagramas de secuencia y diagramas de actividades.

- Secuencia: Modelar cómo interactúan los objetos entre sí para ejecutar una acción (por ejemplo: "Alta de usuario").
- Actividades: Flujos de acciones automáticas o manuales (por ejemplo: proceso de "Configurar un servicio web").

→Diagrama de Clases: Mejorar el diagrama inicial con más atributos, métodos y relaciones

- Añadir herencias, composiciones, dependencias.
- Definir atributos y operaciones importantes.
- Modelar también clases técnicas (por ejemplo, BaseDeDatos, ControladorFTP....)

MATERIALIZAR — Desarrollo del proyecto

→ENTORNOS DE DESARROLLO

- Programar el proyecto usando buenas prácticas de programación.
- Implementar las clases y comportamientos definidos en los diagramas.
- Mantener el repositorio **bien organizado en ramas**:
 - Uso de **pull requests** para integrar ramas al develop.
 - Revisar código (code review) antes de hacer merges.
- **Documentación**:
 - Actualizar diagramas (clases y comportamiento) conforme avanza el proyecto.
 - Comentar el código usando herramientas del entorno de desarrollo (docstrings, Javadoc, XML comments...).

INDICAR LOS ERRORES EN LOS DIAGRAMAS CREADOS ANTERIORMENTE Y COMO LOS HABEIS SOLUCIONADO.

CREAR UNA COPIA DE LOS DIAGRAMAS ANTERIORES , PERO INCLUIR

CIERRE (fase de exposición)

→ENTORNOS DE DESARROLLO

→ Entrega final de Diagramas

Entregar diagramas finales claros, ordenados y completos.

→Entrega de un enlace de GIT con los indicado en la presentación, donde se ve el trabajo realizado de cada usuario a lo largo del tiempo

→Dentro del documento del grupo se debe explicar un LOG con las actividades realizadas de cada usuario y el progreso.