PARTE TEORICA

1-. ¿Qué es un sistema?

R-. Un sistema es un conjunto de hardware y software que trabaja en conjunto para realizar tareas específicas, como computadoras, sistemas operativos y servidores web.

2-. ¿Qué es y qué diferencias tienen una clase abstracta y una clase estática en C#?

R-. Clase abstracta: Se utiliza como una plantilla para heredar y puede contener métodos abstractos y concretos. Puede ser instanciada y heredada.

Clase estática: Contiene miembros estáticos y no se puede instanciar ni heredar. Se usa para agrupar métodos y propiedades compartidos.

3-. ¿Qué es y qué diferencias tienen la herencia y polimorfismo en C#?

R-. Herencia: Clases derivadas heredan propiedades y comportamientos de una clase base.

Polimorfismo: Permite tratar objetos de diferentes clases de manera similar a través de una interfaz común.

4-. ¿Qué es un ciclo de vida del desarrollo de software (SDLC)?

R-. Es un proceso que abarca desde la concepción de un proyecto de software hasta su entrega y mantenimiento, con fases como requisitos, diseño, desarrollo, pruebas, implementación y mantenimiento. Ayuda a gestionar proyectos de software de manera efectiva.

5-. git init: Inicializa un nuevo repositorio Git en un directorio vacío o existente.

git status: Muestra el estado actual de los archivos en el repositorio, indicando qué archivos están pendientes de ser confirmados o modificados.

git add .: Agrega todos los archivos y cambios al área de preparación (staging) para que sean confirmados en el próximo commit.

git commit -m "Mensaje": Confirma los cambios en el repositorio con un mensaje que describe la naturaleza de los cambios realizados.

git log: Muestra un registro de los commits realizados en el repositorio, incluyendo información sobre el autor, la fecha y el mensaje del commit.

git checkout: Se utiliza para cambiar de rama o para restaurar archivos a un estado anterior. En versiones más recientes de Git, se recomienda usar git switch y git restore para estas operaciones, respectivamente.

git checkout -b NombreRama: Crea una nueva rama con el nombre especificado y se cambia a ella al mismo tiempo.

git branch: Muestra una lista de ramas en el repositorio y resalta la rama actual.

git push: Sube los cambios confirmados en tu repositorio local a un repositorio remoto, como GitHub.

git pull: Descarga los cambios más recientes del repositorio remoto y los fusiona con tu rama local.

git merge: Combina una rama con otra, generalmente se utiliza para incorporar cambios de una rama a otra.

git clone: Crea una copia de un repositorio Git remoto en tu sistema local para que puedas trabajar en él.

6-. ¿Cuál es la diferencia entre una metodología tradicional y ágil?

R-. Metodología Tradicional: Enfoque en planificación, requisitos fijos, proceso secuencial, documentación extensa y comunicación formal.

Metodología Ágil: Enfoque en flexibilidad, requisitos cambiantes, desarrollo iterativo, entregables funcionales, comunicación colaborativa y orientación al cliente.

7-. Dar 5 ejemplos de una metodología tradicional y 5 ejemplos de una metodología tradicional ágil

Metodologías Tradicionales:

1-. Modelo en Cascada

2-. Modelo en V

3-. Desarrollo en Espirales

4-. Proceso Unificado (RUP)

5-. Desarrollo Estructurado

Metodologías Ágiles:

1-. Scrum

2-. Kanban

3-. Extreme Programming (XP)

4-. Desarrollo de Software Lean

5-. Desarrollo de Software Crystal

8-. ¿Qué es un Requerimiento Funcional y No Funcional?Requerimientos Funcionales es lo que el sistema debe hacer (funciones específicas).

Requerimientos No Funcionales es cómo el sistema debe hacerlo y sus características (rendimiento, seguridad, usabilidad, etc.).

9-. ¿Qué es SCRUM?

R-. Scrum es un marco de trabajo ágil que se enfoca en la colaboración y la entrega continua de software. Divide el trabajo en sprints cortos, con roles definidos y un backlog priorizado. Se utiliza para proyectos donde los requisitos pueden cambiar y se necesita adaptabilidad.

10-. ¿Cuáles son los roles de SCRUM?

Los tres roles principales en Scrum son:

Scrum Master: Facilita el proceso Scrum y ayuda al equipo a seguir las prácticas de Scrum.

Product Owner: Representa al cliente, prioriza el trabajo y asegura que el producto se alinee con las necesidades del cliente.

Equipo de Desarrollo: Construye el producto y es autoorganizado, con habilidades para completar el trabajo de manera independiente.