Parte teórica

Estudiante: **JorgeEduardo Tapia Tapia**

1. Un sistema es un conjunto de componentes interconectados que trabajan juntos para lograr un objetivo común. Puede ser una colección de hardware y software que desempeña una función específica, como un sistema operativo de una computadora, o una estructura organizativa y procesos diseñados para cumplir una tarea, como un sistema de gestión de pedidos en una empresa.
2. Una clase abstracta en C# es una clase que no se puede instanciar directamente y se utiliza como base para otras clases. En contraste, una clase estática es una clase que no se puede instanciar en absoluto y se utiliza para agrupar métodos y datos relacionados que no requieren una instancia. La principal diferencia es que las clases abstractas permiten la herencia, mientras que las clases estáticas no.
3. La herencia es un mecanismo en C# que permite que una clase adquiera propiedades y métodos de otra clase, lo que fomenta la reutilización de código. El polimorfismo permite que diferentes clases compartan un mismo nombre de método, pero cada una puede implementarlo de manera diferente. La diferencia clave es que la herencia está relacionada con la estructura de las clases, mientras que el polimorfismo se relaciona con el comportamiento de los objetos.
4. El Ciclo de Vida del Desarrollo de Software (SDLC) es un proceso que guía el desarrollo de software desde la concepción hasta la entrega y mantenimiento. Incluye etapas como planificación, diseño, implementación, pruebas y despliegue. Es un marco de trabajo que ayuda a asegurar que un proyecto de software se desarrolle de manera organizada y efectiva.
5. Comandos de Git:

* git init: Inicializa un repositorio Git en un directorio.
* git status: Muestra el estado de los archivos en el repositorio.
* git add .: Agrega los cambios de todos los archivos al área de preparación.
* git commit -m "Mensaje": Guarda los cambios confirmados con un mensaje descriptivo.
* git log: Muestra el historial de confirmaciones.
* git checkout: Cambia entre ramas o puntos de confirmación.
* git checkout -b NombreRama: Crea y cambia a una nueva rama.
* git branch: Muestra la lista de ramas.
* git push: Sube cambios al repositorio remoto.
* git pull: Obtiene cambios del repositorio remoto.
* git merge: Fusiona ramas o cambios.
* git clone: Clona un repositorio remoto en un directorio local.

1. La metodología tradicional se basa en un enfoque secuencial y planificado para el desarrollo de software, mientras que la metodología ágil se centra en la colaboración, la adaptación y la entrega iterativa de software. La metodología tradicional tiende a ser más rígida y requiere una planificación detallada, mientras que la metodología ágil es más flexible y se adapta a los cambios.
2. Ejemplos de metodologías tradicionales: Modelo en Cascada, Modelo en V, Modelo en Espiral, RUP, etc. Ejemplos de metodologías ágiles: Scrum, Kanban, XP (eXtreme Programming), Crystal, Lean, etc.
3. Un Requerimiento Funcional describe lo que debe hacer un sistema o software, como las funciones, características y comportamientos específicos. Un Requerimiento No Funcional se refiere a cómo debe funcionar el sistema, incluyendo aspectos como la seguridad, el rendimiento y la usabilidad, pero no describe una función específica.
4. Scrum es un marco de trabajo ágil ampliamente utilizado para gestionar proyectos de desarrollo de software. Se basa en la colaboración, la transparencia y la adaptación continua. Scrum divide el trabajo en sprints, que son iteraciones cortas, y utiliza roles como el Scrum Master y el Product Owner.
5. Los roles de Scrum incluyen:

* Scrum Master: Facilita el proceso Scrum y ayuda al equipo a eliminar obstáculos.
* Product Owner: Representa las necesidades del cliente y prioriza el trabajo del equipo.
* Equipo de Desarrollo: Realiza el trabajo de desarrollo y entrega el producto.
* Stakeholders: Personas interesadas en el proyecto que aportan sus opiniones y necesidades.