***PARTE TEORICA. -***

***Explicar en sus palabras cada pregunta:***

1. *¿Qué es un sistema?*

Un sistema son varios elementos interrelacionados para realizar un trabajo en específico, Un ejemplo relacionado a la programación es la realización de un sistema de facturación en donde se crearán varios procesos los cuales nos permitirá detallar los datos que estarán en la misma.

Otro ejemplo sería en el Sistema de cables en donde todos estos estarán conectados a algún aparato electrónico para compartir ya sea información o energía.

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

Una clase abstracta puede estar definida sin ningún parámetro dentro.

Una clase estática sí tiene parámetros. Sin embargo, esta no puede ser instanciada en otra clase

1. *¿Qué es y qué diferencias tiene la herencia y polimorfismo en C#?*

En la herencia podemos heredar los atributos de una clase padre a otra clase hijo.

En el polimorfismo podemos usar un método-objeto en varias clases.

1. *¿Qué es un ciclo de vida del desarrollo de software (SDLC)?*

Son las etapas que se deben cumplir para el desarrollo de un software de calidad que consiste desde la idea hasta el mantenimiento y la entrega finalizada del proyecto(software).

* Requisitos.
* Análisis.
* Diseño.
* Implementación.
* Pruebas.

Puntos mencionados por el Ingeniero.

1. *Para qué sirven estos comandos de Git:*

* Git init: Sirve para crear una subcarpeta llamada “git” dentro de la carpeta principal del proyecto y así mismo poder usar los comandos de Git.
* Git status: Sirve para ver el estado del proyecto en Git, incluyendo si los cambios han sido guardados (commiteados) o no.
* Git add .: Sirve para añadir los archivos a la carpeta local (repositorio) y finalizando el guardado con el commit -m “mensaje”.
* Git commit -m “mensaje”: Este comando sirve para confirmar el guardado de los cambios en los archivos dando paso al siguiente commit.
* Git log: Sirve para mostrar el historial de commits en un repositorio.
* Git checkout: Sirve para restaurar archivos desde un commit anterior.
* Git Branch: Este comando se utiliza para crear, mostrar y eliminar ramas en un repositorio.
* Git push: Sirve para enviar los archivos y los cambios confirmados a un repositorio remoto (GitHub).
* Git pull: Sirve para descargar los cambios del repositorio remoto a nuestro repositorio local.
* Git clone: Sirve para copiar un proyecto de un repositorio remoto a un repositorio local (nuestro pc).

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

Las metodologías tradicionales son: Cascada y Rup. Estas metodologías como su nombre lo indican realizan un proceso secuencial pasando por diferentes fases. La metodología que usa la UTO es “Rup”.

La metodología ágil consiste en la entrega temprana de los requerimientos del cliente.

1. Dar 5 ejemplos de una metodología tradicional y 5 ejemplos de una metodología ágil.

De internet:

Metodologías tradicionales:

1. SDLC (Ciclo de vida de desarrollo de software)
2. RUP (Proceso unificado de Rational)
3. Modelo en cascada (Waterfall)
4. Modelo en V
5. Modelo en espiral

Metodologías ágiles:

1. Scrum
2. Kanban
3. XP (Programación extrema)
4. Lean
5. Crystal Clear

***PARTE PRACTICA. -***

**3.- HISTORIAS DE USUARIO**

|  |
| --- |
| HOJA 1: Control del Sistema |
| Como: Gerente General.  Quiero: Tener el control total del sistema.  Para: Controlar a los vendedores, saber en qué turno están trabajando  y monitorear su desempeño diario de ventas. |

|  |
| --- |
| HOJA 2: Reporte de ventas |
| Como: Gerente General.  Quiero: Ver reportes de ventas por día, mes de cada vendedor en el sistema.  Para: Evaluar el desempeño de cada vendedor en un período de tiempo  determinado y tomar decisiones basadas en los datos de ventas. |

|  |
| --- |
| HOJA 3: Registrar y buscar datos del cliente |
| Como: Vendedor.  Quiero: Registrar y buscar los datos del cliente en el sistema.  Para: Acelerar el proceso de emisión de recibos de venta |