**Practica 1**

Brandon Alexander Coca Aguilera

Actualización tecnológica SIS2420 “A”

1. **¿Qué es un sistema?**

Un sistema es un conjunto de elementos relacionados entre si que funcionan como un todo, estos tienen un input, process y output.

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

Una clase abstracta es aquella que no se puede instanciar, solo se puede heredar por otras clases y sirve como base de un comportamiento común para las clases heredadas.

Una clase estática sirve para agrupar métodos o datos que no dependen de ninguna instancia.

La diferencia es que una clase abstracta puede tener miembros no estáticos y una clase estática solo puede tener miembros estáticos.

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

La herencia es la capacidad de una clase para heredar propiedades y métodos de otra clase.

El polimorfismo es la capacidad de una clase pata tener diferentes comportamientos según como se usa.

La diferencia entre estas es que la herencia se refiere a la relación entre clases, mientras que el polimorfismo se refiere al comportamiento de los objetos.

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

Es un proceso de planificación que tiene como objetivo producir un software de calidad, bajo costo y en el menor tiempo posible, satisfaciendo las expectativas y necesidades de los clientes, sus faces son análisis de requisitos, planificación, diseño, desarrollo, pruebas, despliegue y mantenimiento.

1. **Para que sirven estos comandos en Git:**

**Git init:** Crear un nuevo repositorio en Git

**Git status:** Revisar los cambios en el proyecto

**Git add:** Para añadir todos los cambios

**Git commit -m “Mensaje”:** Para guardar los cambios

**Git log:** Muestra todos los commits del repositorio

**Git checkout:** El comando tiene tres funciones: permite recuperar archivos, commits y ramas.

**Git checkout – b NombreRama:** Se usa para crear y cambiar a una rama nueva en un repositorio con un solo comando

**Git branch:** Te muestra las ramas que tienes en el repositorio

**Git push:** Se usa para subir los cambios locales que se han hecho en un repositorio

**Git pull:** Extrae y descarga contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido

**Git merge:** Permite tomar las líneas independientes de desarrollo creadas por git branch e integrarlas en una solo rama.

**Git clone:** Apunta a un repositorio existente y clona o copia dicho repositorio en un nuevo directorio, en otra ubicación.

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

La diferencia entre una metodología tradicional y una ágil es la forma en que se planifica, se ejecuta y se entrega un proyecto. Una metodología tradicional sigue un proceso lineal y secuencial, donde se definen todos los requisitos y el alcance del proyecto al inicio, y se realizan las tareas en orden hasta llegar al producto final. Una metodología ágil sigue en proceso iterativo e incremental, donde se entregan versiones parciales y funcionales del producto en cada ciclo, y se adaptan los requisitos y el alcance de los clientes y los cambios del entorno.

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

Ejemplos de metodología tradicional son: cascada, espiral, prototipado, RUP y MSF.

Ejemplos de metodología ágil son: EssUP, Cascada + Scrum, SXP, SAFe y Agile Kanban.

1. **¿Qué es un Requerimiento Funcional y No Funcional?**

Un requerimiento funcional es una especificación de lo que se debe hacer en un sistema es decir las funciones o características que debe ofrecer a los usuarios.

Un requerimiento no funcional es una especificación de como debe hacerse o comportarse un sistema es decir los atributos o propiedades de calidad que debe cumplir o respetar.

1. **¿Qué es SCRUM?**

Es una metodología ágil para la gestión de proyectos, que se basa en un proceso iterativo e incremental, donde se entregan versiones parciales y funcionales del producto en cada ciclo, llamado sprint. Se apoya en un conjunto de valores, principios y practicas que ayudan a los equipos a estructurar y gestionar su trabajo de forma colaborativa, adaptativa y empírica.

1. **¿Cuáles son los roles de SCRUM?**

Los roles son: El propietario del producto (Product Owner), el equipo de desarrollo (Developers), el scrum master