

Nombre: Brayan Ruben Copa Silas

SIS 24201A

Auxiliar: William Murcio Adhabel Villalpando

Parte Teórica

1. ¿Qué es un sistema?

R: Un sistema es un conjunto de elementos interrelacionados o independientes que trabajan juntos para lograr un objetivo común. Estos elementos pueden ser personas, procesos, información etc.

2. ¿Qué es y que diferencia tienen una clase abstracta y una clase estática en C#?

R: Ambas no se pueden instanciar pero la clase abstracta sirve a otras derivadas pero en cambio las clases estáticas no derivan sino que las funciones que contiene solo pueden ser accedidas desde el nombre.

3. ¿Qué es y que diferencia tienen la herencia y el polimorfismo en C#?

R: La herencia nos sirve para ahorrar código principalmente a través de una clase padre la cual hereda a una clase hijo todos sus atributos y métodos. A diferencia del polimorfismo se refiere a que un método puede tomar varias formas según sea la clase.

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

R: Es un proceso estructurado utilizado en ingeniería de software para planificar, crear, probar y desplegar sistemas de

Sistemas de información y aplicaciones.

Requerimientos
Análisis
Diseño
Implementación
Pruebas

5: Para que sirven estos comandos de GIT

Git init Inicia un nuevo repositorio de Git

" **status** Muestra el estado actual del repositorio, incluyendo archivos sin seguimiento, modificaciones y preparados para la confirmación

" **add** Añade todos archivos modificados al área de preparación para la siguiente confirmación

" **commit -m "Mensaje"** Confirma los cambios en el repositorio. El mensaje proporciona una corta descripción de los cambios efectuados

" **log** Muestra el registro de todas las confirmaciones realizadas en el repositorio

" **checkout** Se utiliza para cambiar entre ramas o para deshacer cambios en los archivos.

" **checkout -b Nombre Rama** Crea una nueva rama con el nombre especificado y cambia a esa rama

" **branch** Muestra una lista de todas las ramas en el

repositorio y la rama actual se resalta

" **push** Envía los cambios locales confirmados al repositorio remoto

" **pull** Obtiene los cambios más recientes del repositorio remoto y los fusiona con el repositorio local

" **merge** Fusiona los cambios de una rama con otra. Se utiliza para combinar ramas

" **clone** Crea una copia local de un repositorio remoto en la máquina del usuario. Se utiliza para obtener una versión completa del repositorio.

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

R: - Las tradicionales tienen un enfoque lineal y secuencial donde cada fase se hace estrictamente además para ello se hace una planificación al principio del proyecto y los cambios no son una opción aconsejable más adelante cuenta con una documentación extensa la diferencia más grande es que puede que la metodología ágil puede realizar cambios se trabaja de manera iterativa e incremental dividiendo el proyecto en pequeñas partes manejables y entregando funcionalidades operativas en ciclos cortos son altamente adaptables a cambios en los requisitos, permitiendo una mayor flexibilidad y capacidad de respuesta a las necesidades del cliente o de suponer que se

documentación es corta, hay entregas frecuentes de manera regular y frecuente, lo que permite al cliente ver progreso constante.

Ejemplos de metodologías tradicional y ágil

Tradicional

- Cascada
- Espiral
- RUP
- Estructurado
- Incremental

Ágil

- SCRUM
- Kanban
- XP (Extreme Programming)
- Crystal Clear
- Modelo Contorno

¿Qué es un requerimiento funcional y No funcional?

R: Un requerimiento no funcional es una especificación que describe características del sistema que no están específicamente relacionadas con las funciones del software o mejor con lo que pide el cliente estos se centran en atributos del sistema como el rendimiento, la usabilidad, la seguridad y otros aspectos de calidad. A diferencia de los requerimientos funcionales, los no funcionales no describen qué hace el sistema, sino cómo lo hace. Un ejemplo la interfaz debe ser intuitiva.

Los requerimientos funcionales es una declaración que describe las funciones específicas que debe realizar un sistema de software. Estos definen la capacidad del

software y lo que el sistema debe hacer o sonar lo que el cliente está pidiendo.

9- ¿Qué es el SCRUM?

R: Es una metodología de desarrollo de software que se basa en mejora constante y en principios iterativos e incrementables, y se centra en la entrega continua de valor al cliente.

10- ¿Roles de SCRUM?

- R: - Product Owner: Representa a los interesados y define las funcionalidades del producto.
- Scrum Master: Facilita el proceso y se asegura que se sigan los prácticas y principios. Asigna tareas.
- Developers: Grupo de profesionales que realiza el trabajo real.

Parte Práctica

1-2-3-4 Esto en archivos adjuntos.

5- Realizar las Historias de Usuario y el Product Backlog para la empresa ChocoMax.

(Actores) Usuario Gerente, Cliente

HU1	Crear Login
Como	Gerente
Quiero	crear cuentas y password
Para	ingresar al sistema

MU2: Gestionar (CRUD) Usuarios

Como: Gerente

Quiero: Crear, leer, actualizar y eliminar (CRUD) de usuarios

Para: administrar las cuentas de usuario

MU3: Generar reportes de ventas

Como: Gerente

Quiero: generar reportes de las ventas por día y mes

Para: controlar el flujo de ingresos

MU4: Gestionar (CRUD) vendedores

Como: Gerente

Quiero: Crear, leer, actualizar y eliminar (CRUD) de vendedores

Para: administrar al personal

MU5: Gestionar (CRUD) clientes

Como: Gerente

Quiero: crear, leer, actualizar y eliminar (CRUD) de clientes

Para: tener un registro de mis clientes

MU6: Registrar recibos

Como: Vendedor

Quiero: realizar recibos sobre las ventas

Para: tener constancia de las ventas

PRODUCT BACKLOG

		Prioridad
1. Crear Login	HU1	1
2. CRUD Usuario	HU2	2
3. CRUD Venecia	HU4	3
4. CRUD Clientes	HU5	4
5. Generar Reportes	HU3	5
6. Generar Recibo	HU6	6

G- Realizar las historias de Usuario y el Product Backlog para mejorar el sistema Dragon FNI

HU1: Registrar paralelos de Laboratorio de la carrera Sistema

Como: Estudiante

Quiero: poder registrarme en horarios accesibles mis laboratorios

Para: no generar choquos con otras materias

HU2: Ver hoja de vida de los docentes

Como: Estudiante

Quiero: ver una buena hoja de vida de los docentes

Para: verificar su preparación y escoger docentes preparados

HU3: Adaptar una plataforma móvil

Como: Estudiante

Quiero: que el dragon FNI cuente con plataforma móvil

Para: utilizar de forma cómoda

NOY: Mejorar el diseño Login

Como: Estudiante

Quiero: contar con un login con mas diseño y atractivo

Para: tener una experiencia mas agradable