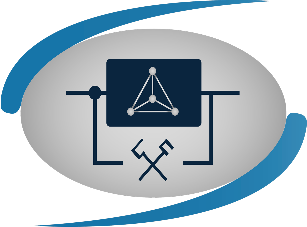
**UNIVERSIDAD TECNICA DE ORURO FACULTAD NACIONAL DE INGENIERIA**

**INGENIERIA DE SISTEMAS E INFORMATICA**

PRACTICA N° 1

**Materia:** Actualización Tecnológica  **Paralelo:** SIS2420 “A”

**Docente:** Ing. Saul Mamani Mamani

**Auxiliar:** Egr. William Mucio Achabal Villalpando

**Nombre:** Univ. Yucra Miranda Denis Albert

# PARTE TEORICA:

## ¿Qué es un sistema?

## Un sistema es un conjunto de elementos que trabajan juntos para lograr un objetivo específico tanto físico como abstracto

## ¿Qué es y qué diferencias tienen una clase abstracta y una clase estática en C#? Clase Abstracta:

## La clase abstracta es servir como clase base para otras clases a las que generalmente se conoce como clases "concretas"

* + No se pueden crear objetos en clase abstracta.
  + Se tiene que heredar para poder utilizar
  + Puede contener métodos sin implementación (abstractos)

## Clase Estática:

La clase estática no se puede instanciar, y todos sus miembros (métodos, propiedades, campos, etc.) deben ser estáticos

* + Contiene métodos y propiedades que pertenecen a la clase en sí.
  + Los miembros se usan a través del nombre de la clase.

## ¿Qué es y qué diferencias tienen la herencia y polimorfismo en C#? La herencia:

La herencia en C# es como tomar las características y comportamientos de una clase existente y utilizarlos para crear una nueva clase. La nueva clase hereda lo que ya existe y puede agregar o cambiar cosas según sea necesario. Esto ayuda a reutilizar código y estructurar clases de manera lógica. Las características de la herencia en C# son:

* + **Jerarquía de clases:** Se pueden crear relaciones lógicas entre clases base y clases derivadas
  + **Reutilización de código:** Permite reutilizar el código de la clase base en la clase derivada
  + **Extensión de funcionalidad:** La clase derivada puede agregar propiedades y métodos heredados

## El polimorfismo:

El polimorfismo es un concepto fundamental en la programación orientada a objetos que se refiere a la capacidad de diferentes objetos de una jerarquía de clases para responder de manera diferente a un mismo mensaje o método. En otras palabras, el polimorfismo permite que objetos de clases distintas puedan ser tratados de manera uniforme a través de una interfaz común.

Métodos virtuales y anulación: En el polimorfismo de tiempo de ejecución, se utilizan métodos virtuales en la clase base y se anulan (override) en las clases derivadas para lograr la respuesta específica de cada clase.

Interfaz Común: Se basa en la definición de una interfaz común o una herencia en la jerarquía de clases para lograr la uniformidad en la forma en que se interactúa con objetos de diferentes tipos.

Resolución en Tiempo de Ejecución: El polimorfismo de tiempo de ejecución permite que la decisión sobre qué método se ejecutará se tome en tiempo de ejecución, según el tipo real del objeto, lo que da lugar a comportamientos específicos de la clase derivada.

“Se puede concluir que la herencia se utiliza para tener una jerarquía entre clases, y el polimorfismo posibilita que objetos pertenecientes a clases distintas reaccionen de maneras diversas ante las llamadas de método"

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

El Ciclo de Vida del Desarrollo de Software (SDLC) es un proceso que guía el desarrollo de software desde la planificación inicial hasta la implementación y el mantenimiento,

asegurando un enfoque organizado y controlado para la creación exitosa de sistemas de software

## Para qué sirven estos comandos de Git:

## Git init: Inicializa un nuevo repositorio Git en un directorio local. Crea un entorno de control de versiones para un proyecto

## Git status: Muestra el estado actual del directorio de trabajo en relación con el repositorio. Indica qué archivos se han modificado, agregado o están pendientes de confirmación.

## Git add.: Agrega todos los archivos modificados en el directorio de trabajo al área de preparación (staging area) para la futura confirmación.

## Git commit –m “Mensaje”: Confirma los cambios en el repositorio con un mensaje descriptivo. Los cambios en el área de preparación se guardan en el historial del repositorio.

## Git log: Muestra un registro de las confirmaciones en el repositorio, incluyendo información sobre el autor, la fecha y el mensaje asociado a cada confirmación.

## Git checkout: Se utiliza para moverte entre ramas o versiones anteriores del proyecto. También se utiliza para crear nuevas ramas si se proporciona un nombre.

## Git checkout – b NombreRama: Crea una nueva rama y te cambia a esa rama. Es una forma abreviada de crear y cambiar a una rama en un solo comando.

## Git branch: Muestra una lista de las ramas existentes en el repositorio. Indica en cuál rama te encuentras actualmente.

**Git push:** Sube los cambios confirmados en tu repositorio local a un repositorio remoto, como GitHub o GitLab. Sincroniza tu trabajo con otros colaboradores

## Git pull: Descarga y fusiona cambios desde un repositorio remoto en tu repositorio local. Actualiza tu versión local del proyecto.

**Git merge:** Fusiona una rama con la rama actual. Combina los cambios de ambas ramas en una sola

**Git clone:** Crea una copia exacta de un repositorio remoto en tu máquina local. Inicializa un nuevo repositorio Git y descarga todo el historial y los archivos del proyecto remoto.

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

## Metodología Tradicional: Es el enfoque de gestión de proyectos secuenciales y planificados con énfasis en la documentación y control rígido.

## Metodología Ágil: Es el enfoque de gestión de proyectos iterativos, flexibles y centrados en la colaboración y la entrega de valor continua.

## Dar 5 ejemplos de una metodología tradicional y 5 ejemplos de una metodología tradicional ágil

**Metodologías Tradicionales :**

1. **Modelo en Cascada (Waterfall):** Un enfoque secuencial donde cada fase (requisitos, diseño, implementación, pruebas, mantenimiento) se completa antes de pasar a la siguiente
2. **Modelo en V:** Un enfoque que extiende el Modelo en Cascada, enfatizando la verificación y validación en cada fase
3. **Modelo en Espiral:** Un modelo iterativo que incorpora elementos de diseño, construcción y prueba en cada ciclo, con énfasis en la gestión de riesgos.
4. **Modelo en Fases Paralelas:** Diferentes fases del proyecto, como diseño y desarrollo, se llevan a cabo en paralelo en lugar de secuencialmente.
5. **Método Rational Unified Process (RUP):** Un marco de desarrollo de software que combina elementos de varias metodologías tradicionales, como el Modelo en Cascada y el Modelo en Espiral.

## Metodologías Ágiles:

1. **Scrum:** Un método donde se trabaja en pequeños bloques de tiempo para lograr objetivos rápidos y adaptarse a los cambios.
2. **Kanban:** Un sistema visual que ayuda a los equipos a gestionar su trabajo y a avanzar de manera constante.
3. **Extreme Programming (XP):** Un enfoque que pone énfasis en la colaboración y la calidad del código.
4. **Lean Software Development:** Un método que busca eliminar el desperdicio y mejorar la eficiencia en el desarrollo de software.
5. **Dynamic Systems Development Method (DSDM):** Se centra en la entrega oportuna y la colaboración activa entre equipos.
6. **¿Qué es un Requerimiento Funcional y No Funcional?**

**Requisitos Funcionales:** Son las funciones y comportamientos específicos que un sistema de software debe realizar, describiendo qué hace el sistema en respuesta a ciertas acciones o entradas.

**Requisitos No Funcionales:** Son las características de calidad, restricciones y limitaciones que debe cumplir un sistema, como el rendimiento, seguridad, usabilidad, escalabilidad

## ¿Qué es SCRUM?

## Es una metodología ágil , enfocada a trabajo en equipo

## ¿Cuáles son los roles de SCRUM? Product Owner (Dueño del Producto):

## Scrum Master (Maestro Scrum):

## Developers (Equipo de Desarrollo):

# PARTE PRACTICA:

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

La empresa ChocoMax está ubicada en la ciudad de Tarija, donde la empresa se dedica a la elaboración y venta de chocolates. Actualmente la empresa gestiona la venta de los chocolates de forma manuscrita, también se detectó que no cuenta con un buen control de los vendedores en que turno están o cuanto fue su venta en el día, lo cual genero perdidas económicas,

En la empresa ChocoMax existen dos turnos (turno mañana y turno tarde), cada vendedor trabaja solamente un turno. El Gerente general sólo le interesa la parte de reportes de las ventas por día, por mes y por vendedor, el vendedor requiere registrar las ventas, buscar o añadir los datos del cliente para emitir un recibo de la venta.

## HU 1: inicio de sesión

* + **Como** gerente
  + **Quiero** acceder al sistema de ventas.
  + **Para** tener un mejor seguimiento de las ventas realizadas de la empresa

## HU 2: Registro de Ventas

* + **Como** gerente
  + **Quiero** ver las ventas realizadas en el día, mes y por vendedor.
  + **Para** para saber que producto es el mas vendido y que vendedor vende mas

## HU 3: Reporte excepcional

* + **Como** gerente
  + **Quiero** recibir informes de ventas grades
  + **Para** ver la excepcionalidad de los vendedores

## HU 4: Reporte problemas

* + **Como** gerente
  + **Quiero** recibir informes de problemas con clientes o con vendedores
  + **Para** solucionar el problema de la mejor forma

## HU 5: Registrar ventas

* + **Como** vendedor
  + **Quiero** que el sistema sea mas agradable
  + **Para** registrar las ventas con mas facilidad

## HU 6: Registrar datos del cliente

* + **Como** Administrador
  + **Quiero** que el sistema pueda agregar los datos del cliente
  + **Para** poder generar recibos y tener un registro del comprador

## HU 7: Registro turno

* + **Como** vendedor.
  + **Quiero** tener un registro de entrada y salida de mi turno
  + **Para** no tener problemas con la empresa respecto a mis horarios

## HU 8: gestionar cuentas

* + **Como** administrador
  + **Quiero** que el vendedor tenga acceso al sistemacon su propia contraseña y usuario
  + **Para** tener una seguridad de que solo miembros de la empresa tengan acceso al sistema

## HU 9:

* + **Como** Administrador
  + **Quiero** dar recibos de venta en papel a todos los compradores
  + **Para** que tengan comprobantes de venta

## Realizar las Historias de Usuario y el Product Backlog para mejorar el Sistema Dragón FNI

HU **1:**

* + **Como** Universitario.
  + **Quiero** tener acceso a la información nueva sobre cambios de materia y horarios
  + **Para** poder estar al tanto y aprovechar de mejor manera los nuevos cursos

## HU 2:

* + **Como** universitario.
  + **Quiero** ver los docentes disponibles
  + **Para** poderescoger a la mejor opción

## HU 3:

* + **Como** administrador
  + **Quiero** los datos actualizados del estudiante como (teléfono celular )
  + **Para** facilitar comunicarse con el estúdiate

## HU 5:

* + **Como** administrador
  + **Quiero** que los docentes generen notas de formas mas eficientes en el sistema
  + **Para** que el estudiante este informado lo mas antes posible su calificación

## HU 6:

* + **Como** usuario
  + **Quiero** una notificación de eventos de suma importancia de manera automática
  + **Para** no tener problemas con el tiempo

## HU 7:

* + **Como** universitario
  + **Quiero** las calificaciones de semestres pasados y con que ingenieros pase clases
  + **Para** saber que ingeniero fue mas accesible en su calificación