PRACTICA N° 1

**Materia:** Actualización Tecnológica SIS2420 “A” **Semestre:** II/2023

**Docente**: Ing. Saul Mamani M. **Fecha de Emisión:** 23/09/2023

**Auxiliar:** Egr. William Mucio Achabal Villalpando **Fecha de Entrega:** Primer Parcial

# Parte Teórica 30 pts.

**EXPLICAR EN SUS PALABRAS CADA PREGUNTA**

1. ¿Qué es un sistema?

Es un conjunto de elementos interrelacionados para cumplir una finalidad determinada.

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

* **Clase Abstracta:** Es una clase que no se puede instanciar mediante objetos, implementa atributos y comportamientos comunes a otras clases, de esta forma una clase abstracta es usada como una especie de platilla para crear clases derivadas con características similares que heredaran la clase abstracta para implementar sus métodos abstractos y sus métodos concretos.
* **Clase Estática**: Es una clase que no se puede heredar. Tiene métodos estáticos que pueden ser usados directamente desde la definición de la clase.
* **Diferencia:** Lo único en común entre ambos tipos de clases es que no se pueden crear instancias de ninguna de ellas, ósea que no se pueden crear objetos de estas clases, más allá de ello sus diferencias recaen en el uso y propósito, las clases abstractas se implementan como plantillas para generar clases con métodos similares, por lo que para usar una clase abstracta deben de heredarse sus características a clases derivadas, mientras que en el caso de las clases estáticas, éstas no pueden heredarse ni derivarse en subclases, se implementan para generar funcionalidades que requieran usarse de forma directa desde una clase donde se pueda copilar todas las funcionalidades que se requieran, como una especie de caja de herramientas.

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

* **Herencia:** Es un mecanismo que implementa la propiedad de la generalización para definir nuevas clases a partir de clases base, estas clases derivadas además de heredar las características y el comportamiento de las clases base también tendrán sus características propias.
* **Polimorfismo:** Es una propiedad que permite a un operador o función actuar de forma distinta en función del objeto que lo invoca, prácticamente es tener una función definida en varias clases con el mismo nombre pero que realizan diferentes acciones en cada una de estas.
* **Diferencias:** Ambos conceptos van de la mano ya que la herencia permite el polimorfismo de aquellos métodos pertenecientes a clases bases que actuaran de forma distinta en cada clase derivada para corresponder al contexto de estas, sus diferencias se remarcan en sus propias definiciones y propósitos, la herencia se centra en la relación entre clases mediante una jerarquía de superclases con subclases que comparten características, mientras que el polimorfismo se centra en permitir distintos comportamientos de un mismo método en función de la clase que la define, esto en tiempo de ejecución.

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

Es el proceso que describe las etapas por las que atraviesa un software desde su desarrollo, mantenimiento hasta llegar al final de su vida útil. Estas etapas son:

* **Requerimientos:** Etapa donde se registran y documentan los requerimientos del software.
* **Análisis:** Etapa donde se determina lo que hará el software.
* **Diseño:** Etapa que determina como se construirá el software, complementando la etapa con algunas herramientas para ayudar al proceso del diseño.
* **Implementación:** Etapa donde se comienza con el desarrollo del software mediante la programación de esta, hasta llegar al producto final.
* **Pruebas:** Se realizan pruebas al producto final para asegurar el correcto funcionamiento y verificar si se implementaron correctamente los requerimientos establecidos.
* **Mantenimiento:** Etapa donde se continua con la mejora del software, junto a la solución de errores, se mantiene al software en funcionamiento y se dispone la posibilidad a adaptaciones según las necesidades del usuario.

1. *Para qué sirven estos comandos de Git:

Git init - Inicializa un repositorio Git en el directorio actual.

**Git status - Muestra el estado actual del repositorio.

Git add . - Agrega todos los cambios en el directorio de trabajo al área de preparación.

**Git commit -m “Mensaje” - Confirma los cambios con un mensaje descriptive.

Git log - Muestra un registro de confirmaciones.

*Git checkout - Cambia a una rama específica.

**Git checkout - b NombreRama - Crea una nueva rama con el nombre especificado y cambia a esa rama.

Git branch - Lista las ramas locales y resalta la actual.

****Git push - Sube los cambios locales confirmados a un reposito remoto.

Git pull - Obtiene y fusiona cambios desde un repositorio remoto.

Git merge - Fusiona los cambios de una rama en la rama actual.

Git clone - Crea una copia local de un repositorio remoto.

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

* **Diferencias. –** Sus diferencias recaen en sus enfoques a la hora de realizar el proyecto, la metodología tradicional es secuencial y detallada, sigue una serie de pasos en un orden secuencial de principio a fin, por lo que un requisito es empezar un proyecto con una extensa y detallada planeación antes de comenzar el desarrollo haciéndolo una metodología excelente para proyectos grandes y bien definidas. Una metodología ágil se enfoca en el producto, para ello se enfoca en acelerar el proceso de forma eficiente, implementando una mecánica de desarrollo iterativo e incremental, donde se forma por etapas iterativas con posibilidad de retroalimentación en cada iteración, haciéndolo un proceso flexible a la adaptación, el producto se va formando en partes pequeñas pero funcionales hasta formar el producto final haciéndolo una metodología excelente para proyectos pequeños o medianos sujetos a constantes cambios.

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

**Metodologías Tradicionales**

* **Espiral**

Permite el control preciso de recursos y riesgos del proyecto.

Permisiva con cambios y participación activa del cliente.

Permite un proceso iterativo para prolongar la duración del proyecto.

No adecuada para proyectos pequeños.

* **Cascada**

Enfocada en generar un proyecto bien estructurado y documentado.

Imposible de fragmentar proyectos complejos.

No permite la retroalimentación por parte del cliente.

* **Prototipo**

No requiere requerimientos robustos o estables al inicio.

Promueve un acercamiento directo a las necesidades del cliente.

Facilita la presentación de resultados al cliente.

Suele confundirse el prototipo con el producto final

* **Incremental**

Permite cambios de requerimientos en cada iteración.

Exige una planificación exhaustiva.

Requiere requerimientos bien definidos.

* **Desarrollo rápido de aplicaciones (RAD)**

Flexible y adaptable a cambios.

Fácil de trasladar a otros entornos.

Requiere tiempos de entrega precisos.

No se recomienda para proyectos pequeños.

Requiere de expertos en diseño y programación.

**Metodologías Ágiles**

* **Programación Extrema (XP)**

Constantes pruebas que garantizan la calidad del código.

Implica una activa participación del cliente.

Fuerte dependencia en el equipo.

Dificultad en generación de documentación.

* **Kanban:**

Supervisión constant del rendimiento.

Tiempos de producción estables.

La gestión afecta tiempos.

Trabajr con recursos limitados puede causar problemas.

* **Scrum:**

Retroalimentación constante del equipo y clientes.

Calendarios de supervisión definidos.

Clasificación de actividades por importancia.

Los participantes deben ser expertos.

No recomendable para proyectos grandes.

* **Lean:**

Optimización de costos y velocidad de entrega.

Eliminación de elementos sin valor.

Equipo motivado y colaborativo.

Requiere expertos en todos los ámbitos.

Dependencia del equipo.

* **Desarrollo basado en Características (FDD):**

Eliminación de elementos sin valor.

Cumplimiento de requerimientos.

Eficiente control del tiempo y el equipo.

Necesita líder experto en programación.

Documentación casi inexistente.

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

* **Requerimiento Funcional. -**  Son las funcionalidades que se piden por parte del cliente que el sistema debe realizar de forma explícita.
* **Requerimiento No Funcional. –** Son las funcionalidades que un sistema debe adoptar en términos de seguridad, rendimiento, flexibilidad y portabilidad, estas funcionalidades no suelen concernir al cliente.

1. ¿Qué es SCRUM?

* **Definición:** Es una metodología de trabajo ágil de gestión de proyectos en base al trabajo en equipo organizado mediante roles y con periodos laborales basado en iteraciones.

1. ¿Cuáles son los roles de SCRUM?

* **Roles:** 
  + **Product Owner (PO):** Su rol en el equipo es representar los intereses del cliente (stakeholder), para ello trabaja y organiza lo que es el Product Backlog donde se copilara todos los requerimientos del cliente con respecto a las funcionalidades del proyecto. A lo largo del desarrollo del proyecto actuara como un intermediario entre el equipo y los stakeholders.
  + **Scrum Master (SM):** Su rol en el equipo es el de un líder encargado de la efectividad del equipo Scrum ayudando al equipo a comprender la teoría y práctica de Scrum.
  + **Developers:** Su rol en el equipo implica llevar a cabo el armado del proyecto, en función del Product Backlog, trabajan creando un Increment en cada iteración hasta formar el producto final, durante el proceso están en constante comunicación con el PO y el SM, son autosuficientes y adaptativos para confrontar los impedimentos que se presenten.

## Parte Práctica 70 pts.

1. **Realizar un programa utilizando una clase estática que permita ingresar un número por teclado y te muestre en su parte literal.**

**Entrada**

Numero: 55

**Salida**

Cincuenta y cinco

## Realizar un programa utilizando listas que te permita ingresar n números por teclado donde cada número entre en las siguientes listas:

**Entrada**

¿Cuantos números deseas añadir?

* + 7

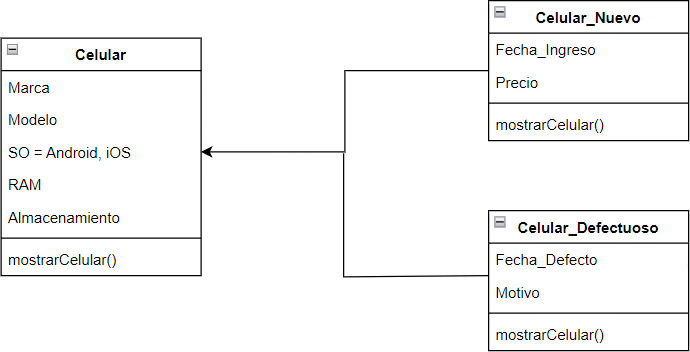
Añada 7 números:

* + 2, 3, 5, 10, 5, 4, 6

**Salida**

* + Lista 1: 2, 4, 6,10 (Múltiplos de 2)
  + Lista 2: 2, 3, 5 (Primos)
  + Lista 3: 5, 10 (Múltiplos de 5)
  + Lista 4: 6 (Perfectos)

## Realizar un programa que tenga las siguientes clases utilizando polimorfismo y herencia La clase Celular debe ser una clase abstracta



1. **Del ejercicio 3 crear una lista utilizando la clase Celular\_Nuevo**
   * Añadir 10 celulares\_nuevos.
   * Crear una función Prom\_Celular, utilizando expresiones lambda sacar el promedio del precio de los celulares.
   * Crear una función Cel\_MarcaS para buscar los celulares de Marca = Samsung, utilizando expresiones lambda.
   * Crear una función Celular\_RSA, utilizando consultas LinQ mostrar los celulares que son de RAM = 8GB, SO = Android y Almacenamiento de 128 GB.
   * Crear una función Celular\_Ingreso, utilizando consultas LinQ mostrar los celulares que ingresaron el año 2005.
   * Crear dos funciones, la primera función usando: Expresiones lambda y la segunda función: consultas LinQ, donde se debe mostrar el modelo y el precio de los celulares Apple.

## 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.

# HISTORIAS DE USUARIO.

|  |  |
| --- | --- |
| **HU1: Iniciar Sesión** | |
| **Como** | Gerente General |
| **Quiero** | Una validación de usuario |
| **Para** | Ingresar al sistema de información web |

|  |  |
| --- | --- |
| **HU2: Gestionar (CRUD) Usuarios** | |
| **Como** | Gerente General |
| **Quiero** | Crear, Leer, Editar y Eliminar (CRUD) usuarios |
| **Para** | Controlar las acciones que realizan |

|  |  |
| --- | --- |
| **HU3: Gestionar (CRUD) Vendedores** | |
| **Como** | Gerente General |
| **Quiero** | Crear, Leer, Editar y Eliminar (CRUD) vendedores |
| **Para** | Tener un gestión de los vendedores en sus correspondientes turnos |

|  |  |
| --- | --- |
| **HU4: Reportar ventas** | |
| **Como** | Gerente General |
| **Quiero** | Generar un reporte de las ventas por día, mes, y vendedor. |
| **Para** | Tener un control de las ventas realizadas en determinado periodo. |

|  |  |
| --- | --- |
| **HU5: Registrar Ventas** | |
| **Como** | Vendedor |
| **Quiero** | Registras las ventas realizadas en mi turno |
| **Para** | Tener constancia de mis ventas hechas durante mi turno. |

|  |  |
| --- | --- |
| **HU6: Gestionar (CRUD) Clientes** | |
| **Como** | Cajero |
| **Quiero** | Crear, Leer, Editar y Eliminar (CRUD) clientes |
| **Para** | Acceder a sus datos para cada venta |

|  |  |
| --- | --- |
| **HU7: Emitir Recibo** | |
| **Como** | Vendedor |
| **Quiero** | Emitir recibo de todos los productos vendido por cliente. |
| **Para** | Que el cliente tenga constancia de su consumo en nuestro negocio. |

# PRODUCT BACKLOG.

|  |  |
| --- | --- |
| **Historia de**  **usuario** | **Descripción** |
| **HU1** | **Iniciar Sesión** |
| **HU2** | **Gestionar (CRUD) Usuarios** |
| **HU3** | **Gestionar (CRUD) Vendedores** |
| **HU4** | **Reportar Ventas** |
| **HU5** | **Registrar Ventas** |
| **HU6** | **Gestionar (CRUD) Clientes** |
| **HU7** | **Emitir Recibo** |

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

|  |  |
| --- | --- |
| **HU1:Acceder a Autobuses** | |
| **Como** | Estudiante |
| **Quiero** | Tener acceso a los horarios y rutas de los autobuses de la facultad. |
| **Para** | Poder planificar y optimizar mis desplazamientos de manera eficiente. |

|  |  |
| --- | --- |
| **HU2: Acceder a herramientas virtuales** | |
| **Como** | Estudiante |
| **Quiero** | Acceder a una lista que muestre todas las herramientas virutales que ofrece la facultad. |
| **Para** | Acceder de manera rápida y conveniente a las herramientas. |

|  |  |
| --- | --- |
| **HU3: Anotar notas parciales** | |
| **Como** | Docente |
| **Quiero** | Registrar y mostrar notas parciales a los estudiantes a lo largo del semestre. |
| **Para** | Mantener a los estudiantes informados sobre su desempeño académico a lo largo del semestre y proporcionar retroalimentación. |

|  |  |
| --- | --- |
| **HU4: Acceder a información** | |
| **Como** | Estudiante |
| **Quiero** | Acceder a instrucciones y detalles que describan sobre los pasos para realizar cualquier trámite estudiantil disponible en la facultad. |
| **Para** | Facilitar la realización de trámites. |

|  |  |
| --- | --- |
| **HU5: Acceder a lista docentes** | |
| **Como** | Estudiante |
| **Quiero** | Acceder a un listado de docentes activos en el semestre, organizados por materias. |
| **Para** | Tomar decisiones informadas sobre planificación y elección de paralelos en función del docente. |

|  |  |
| --- | --- |
| **HU6: Acceder al personal** | |
| **Como** | Estudiante |
| **Quiero** | Acceder a información acerca del personal a cargo de la administración de la facultad, incluyendo el centro de estudiantes. |
| **Para** | Estar informado acerca de quienes manejan la facultad. |

|  |  |
| --- | --- |
| **HU7: Acceder a listado auxiliares** | |
| **Como** | Estudiante |
| **Quiero** | Acceder a un listado actualizado de los auxiliares actuales del semestre. |
| **Para** | Facilitar la comunicación y búsqueda mediante la información de contacto de los auxiliares. |

**Nota: Subir toda la Práctica en un repositorio público de su cuenta GitHub y enviar el enlace del repositorio al inbox por Slack, si entregas la práctica antes del parcial obtendrás +10 puntos a la nota obtenida. ¡Prácticas iguales serán anuladas!**

Vo. Bo. Ing. Saul Mamani M. Egr. William Mucio Achabal Villalpando

# Docente Auxiliar