

INVESTIGACION 30% LISTA DE COTEJO

NOMBRE DEL DOCENTE: María de los Ángeles Pelayo Vaquero		FIRMA DEL DOCENTE	
DATOS GENERALES DEL PROCESO DE EVALUACIÓN			
NOMBRE DEL ALUMNO: MONTAN MARTINEZ ANNETTE			GRUPO 310-A
PRODUCTO: Investigación	UNIDAD: 4	FECHA 13-DIC- 2023	PERIODO ESCOLAR: SEPTIEMBRE 2023 – ENERO 2024

INDICADOR	VALOR	PORCENTAJE OBTENIDO
Presentación - Formato	5	5
Introducción Idea clara del contenido del trabajo, motivando al lector a continuar con su lectura y revisión	5	3
Desarrolla el objetivo	5	3
Desarrollo de la investigación La investigación cumple con el tema solicitado	5	3
Desarrolla la conclusión de investigación	5	3
Gramática y ortografía	2	5
Bibliografía	3	2
Total	30%	24%

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS
TUXTLA



CARRERA
INGENIERÍA EN INFORMÁTICA



MATERIA
FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN

INVESTIGACION-MODELADO

UNIDAD 4

DOCENTE
MARIA DE LOS ANGELES PELAYO VAQUERO

ALUMNA:
ANNETTE MONTAN MARTÍNEZ

GRUPO: 310-A

SAN ANDRÉS TUXTLA, VER. 12 DE DICIEMBRE DEL 2023

DIAGRAMA CASO (S) DE USO

Un diagrama de casos de uso es una representación gráfica que muestra las interacciones entre diferentes actores (usuarios o sistemas) y un sistema en particular. En el caso de una tienda de ropa, los actores pueden incluir clientes, vendedor y administradores del sistema.

- ❖ El Cliente puede realizar acciones como buscar productos, ver detalles, realizar compras.
- ❖ El vendedor puede realizar acciones similares a las del Cliente, pero también puede gestionar el inventario y las ventas en la ropa.
- ❖ Administrador tiene acceso a funciones más administrativas, como gestionar las ventas y generar informes.

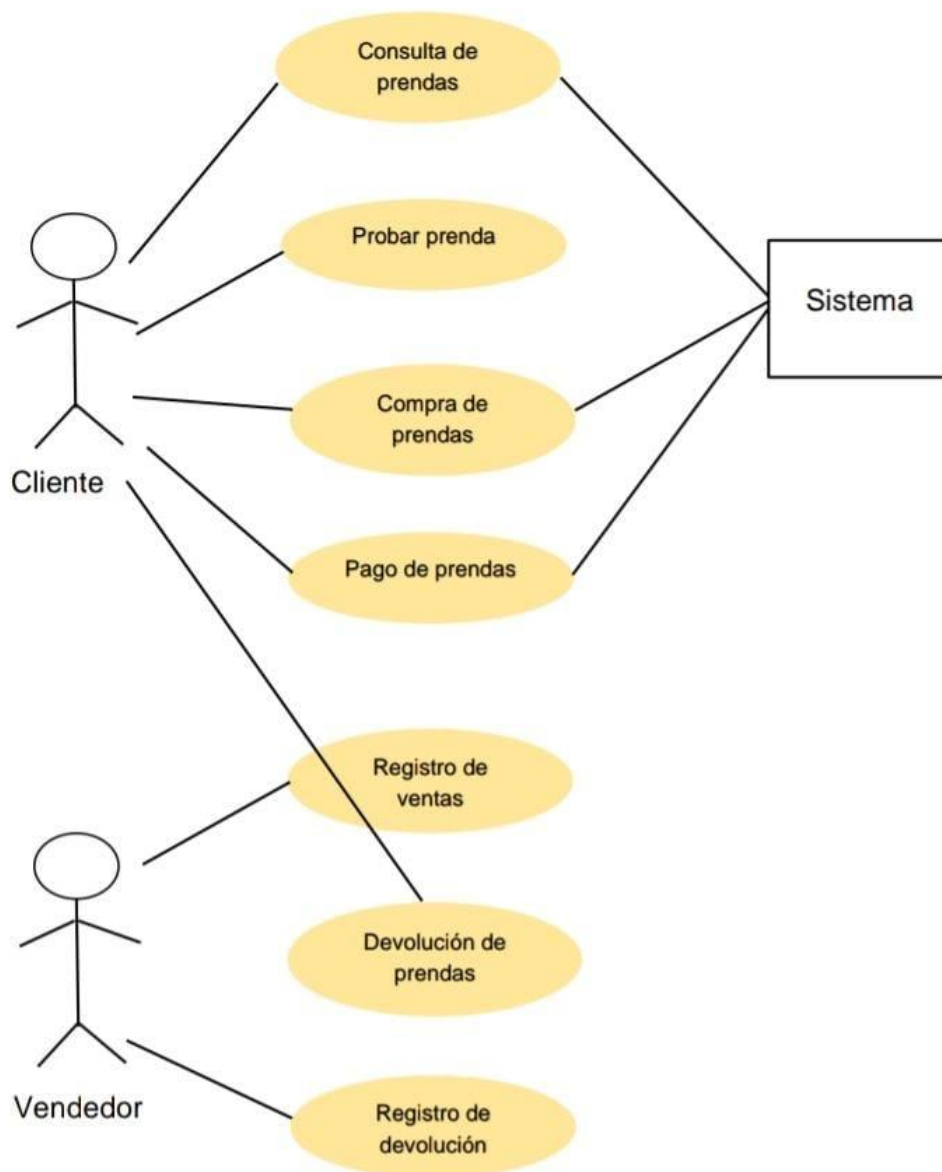


DIAGRAMA DE SECUENCIA

El diagrama de secuencia es una representación visual de cómo interactúan diferentes objetos en un sistema a lo largo del tiempo. Este es un ejemplo se basa según las interacciones específicas que ocurren en la tienda de ropa.

- ❖ El cliente selecciona la prenda.
- ❖ El sistema muestra los detalles.
- ❖ El cliente agrega el producto.
- ❖ El cliente liquida la venta
- ❖ El cajero solicita forma de pago.
- ❖ El cliente hace el pago.

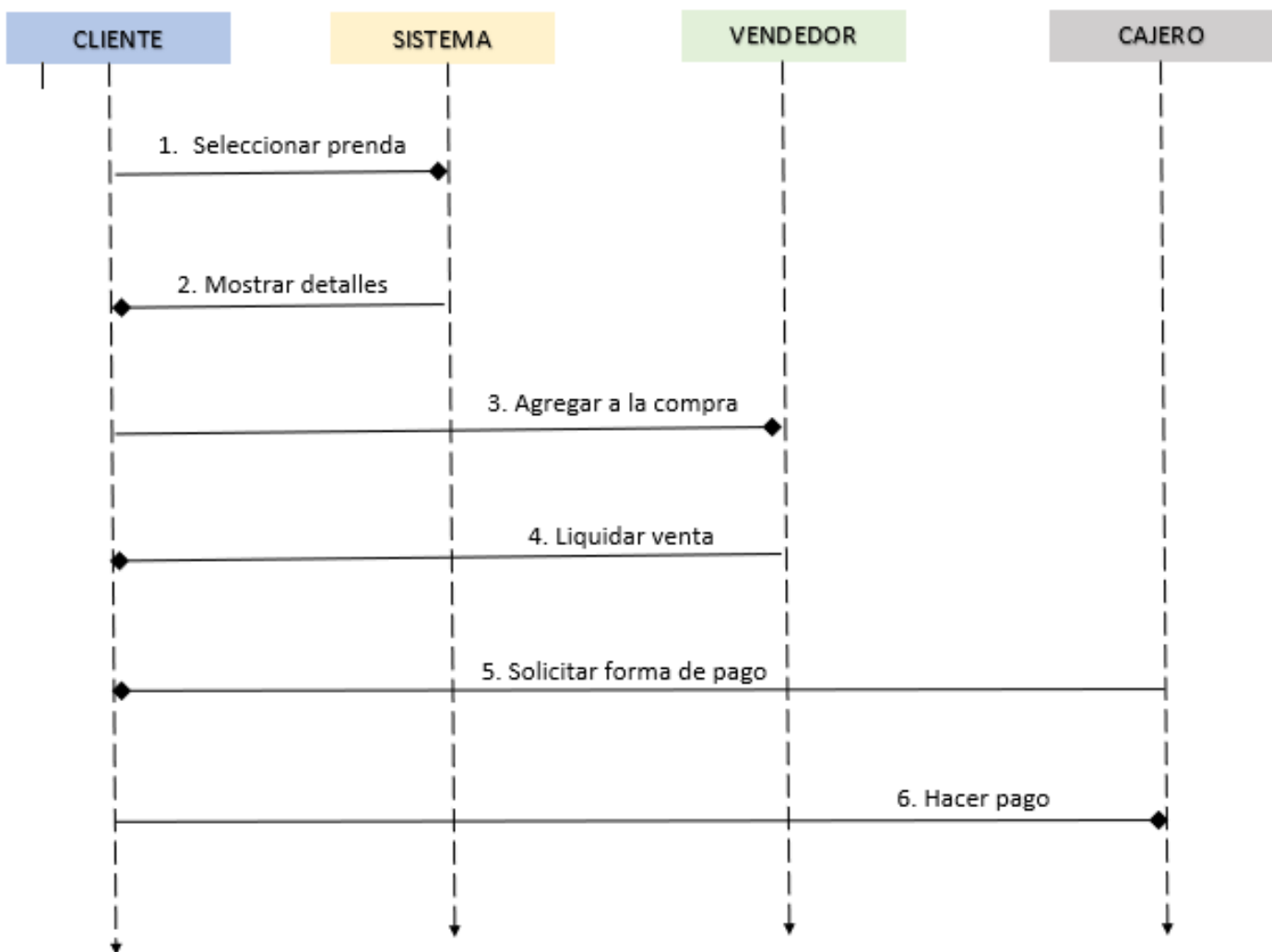


DIAGRAMA DE CLASE

- ❖ Tienda de ropa: Representa la clase principal que tiene información sobre la tienda, como su nombre, dirección, teléfono y número de empleados.
- ❖ Producto: Representa la clase que modela los productos que la tienda vende. Tiene atributos como el ID, nombre, precio, cantidad y fecha.
- ❖ Cliente: Representa la clase que modela a los clientes de la tienda. Tiene atributos como el ID, nombre, correo y teléfono.
- ❖ Venta: Representa la clase que modela a la venta de la tienda. Tiene atributos como el ID de venta, ID de producto, cantidad, fecha, ID del cliente, total.

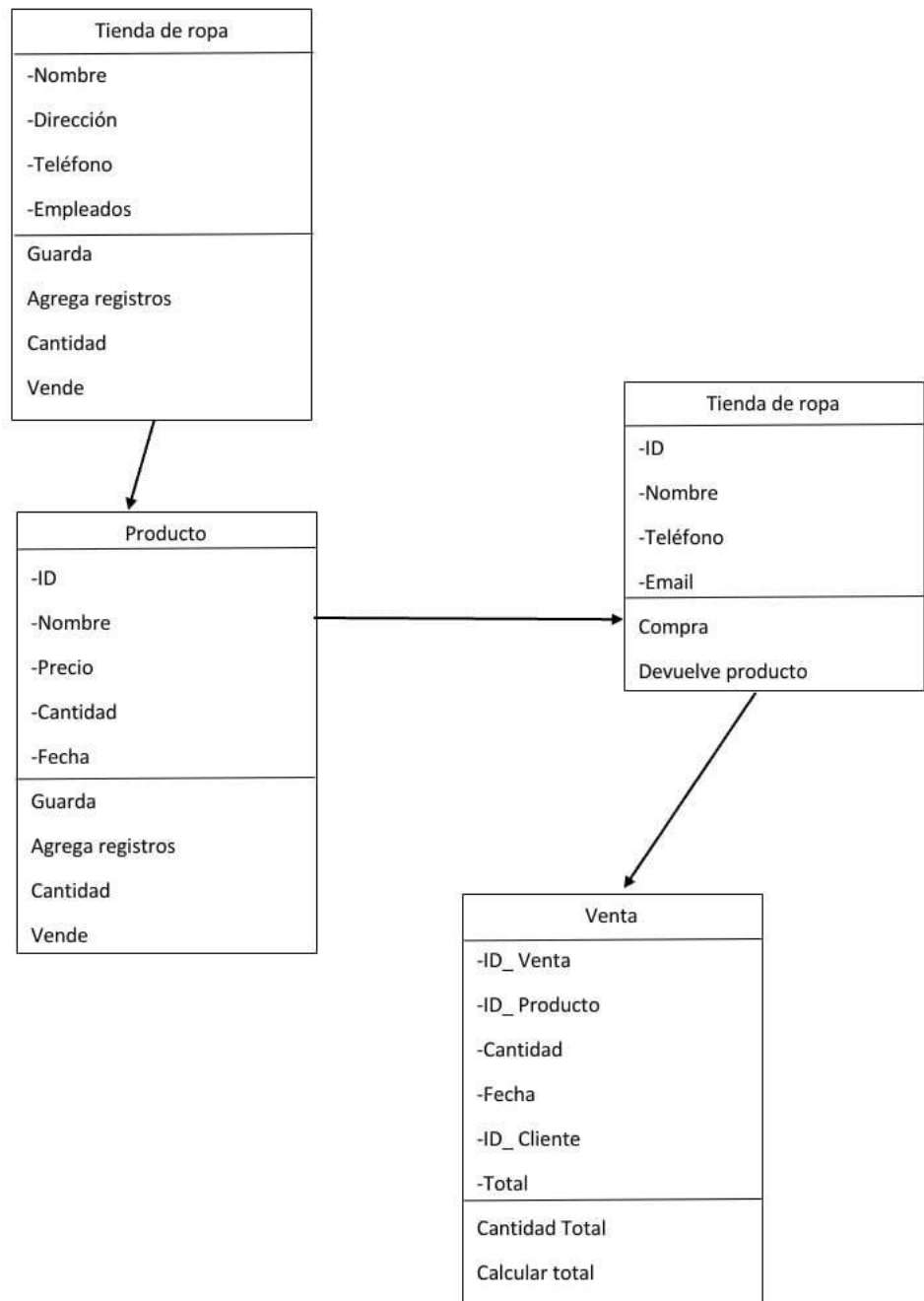
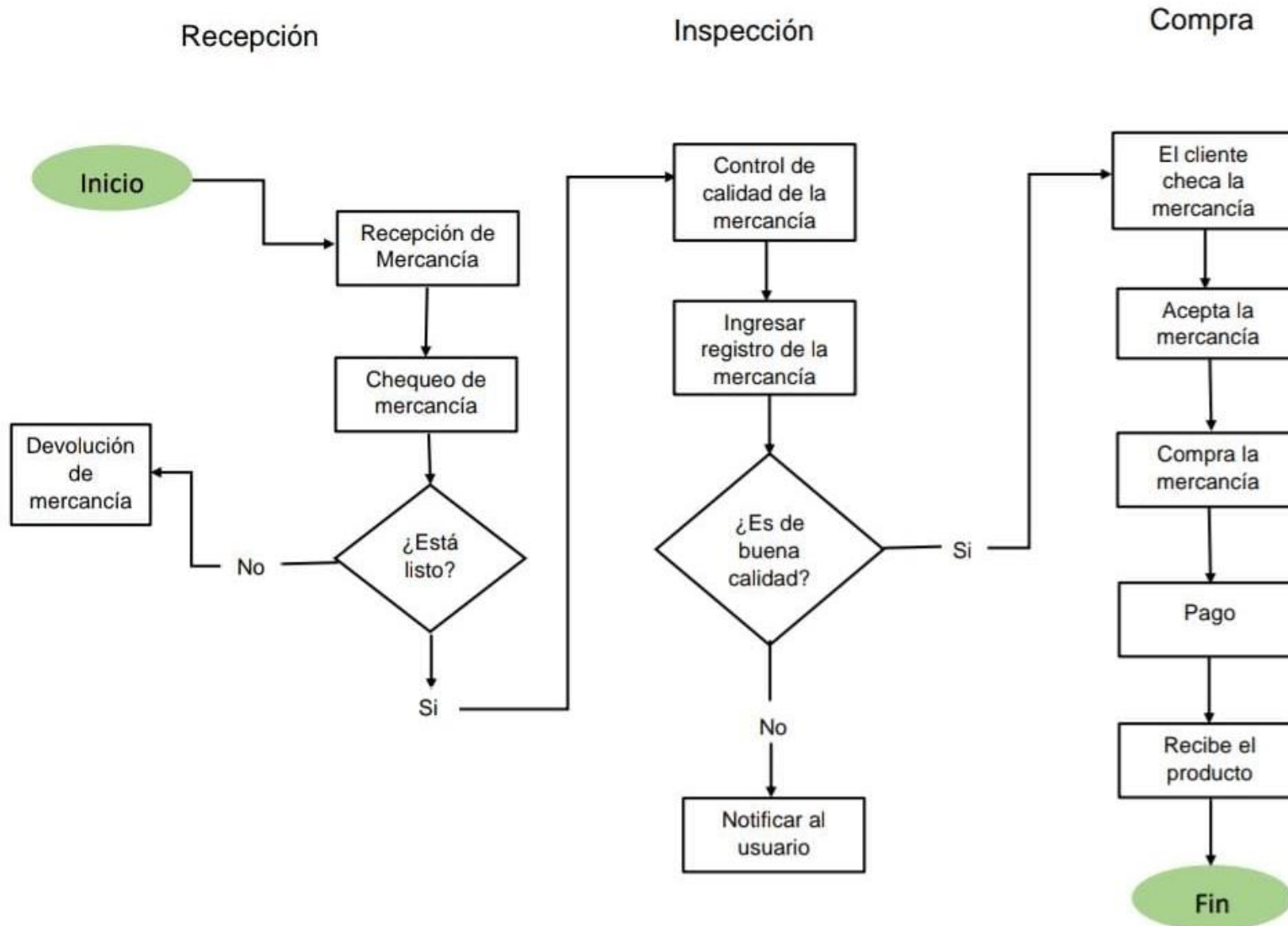


DIAGRAMA DE ACTIVIDADES

Cada rectángulo en el diagrama representa una actividad, y las flechas indican el flujo de control de una actividad a otra. Es una representación básica y se puede expandir las necesidades específicas de la tienda de ropa



INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS
TUXTLA



CARRERA
INGENIERÍA EN INFORMÁTICA



MATERIA
FUNDAMENTOS DE SISTEMAS DE INFORMACION

UNIDAD 4

ENSAYO

DOCENTE
MARIA DE LOS ANGELES PELAYO VAQUERO

ALUMNA:
ANNETTE MONTAN MARTÍNEZ

GRUPO: 310-A

SAN ANDRÉS TUXTLA, VER. 30 DE NOVIEMBRE DEL 2023

SEPTIEMBRE-ENERO 2023-2024

INTRODUCCIÓN

Hace décadas, cuando la informática estaba en sus primeros pasos, los programadores se enfrentaban a la tarea de dar vida a ideas a través de líneas de código. Este desafío llevó al surgimiento de diferentes enfoques de programación, cada uno con su filosofía única para abordar la creación de software. Desde el enfoque estructurado hasta los paradigmas de vanguardia, la programación ha experimentado una evolución fascinante.

El enfoque estructurado, pionero en su tiempo, abogaba por la división de los programas en módulos lógicos y coherentes. Esta metodología facilitó la comprensión y el mantenimiento del código, especialmente en proyectos de tamaño moderado. Sin embargo, a medida que las aplicaciones crecían en complejidad, surgieron desafíos.

Esta ingeniería trata con áreas muy diversas de la informática y de las Ciencias de la Computación, tales como construcción de compiladores, Sistemas Operativos, o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de Sistema de Información y aplicables a infinidad de áreas (negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet Intranet, etc.).

EL ENFOQUE ESTRUCTURADO

En el Enfoque Estructurado se usan los DFD (Diagrama de Flujo de Datos) como principal herramienta para entender al sistema antes de plasmarlo a código fuente. DFD es un diagrama en el que participan procesos (métodos), flujo de datos (argumentos) y archivos (Base de datos). Hay de diferentes niveles dependiendo la complejidad del sistema que se analiza, hablando de lenguajes tiene mucha diferencia con la orientada a objetos, un mínimo cambio en el código puede llegar alterar al resto del programa cosa que en la orientada a objetos eso no sucede lo cual es una ventaja porque así no se pierde tiempo en arreglar cosas ya hechas. Una desventaja es que una porción de código en lenguaje estructurado es difícil que pueda servir en otros proyectos, esto si es habitual en lenguaje orientada a objetos, con solo importar clases ya hechas se escribe menos código y se ahorra tiempo.

Diagrama de Flujo de Datos

- Un diagrama de flujo de datos (DFD) es un modelo lógico-gráfico para representar el funcionamiento de un sistema en un proyecto software.

Diccionario de Datos

- El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema
- El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

Diseño de Módulo

Un modelo de datos es un lenguaje orientado a describir una Base de Datos. Típicamente un Modelo de Datos permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- Operaciones de manipulación de los datos: Operaciones de agregado, borrado, modificación y recuperación de los datos de la base.
- Otro enfoque es pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

Proceso

Conjunto de tareas lógicamente relacionadas que existen para obtener un resultado definido dentro de un negocio o proyecto.

EL ENFOQUE ORIENTADO A OBJETOS

Un objeto es aquello que tiene estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos). La estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancian y objeto son intercambiables. Una clase es un conjunto de objetos que comparten una estructura y comportamiento común.

La diferencia entre un objeto y una clase es que un objeto es una entidad concreta que existe en tiempo y espacio, mientras que una clase representa una abstracción, la «esencia» de un objeto, tal como son. De aquí que un objeto no es una clase, sin embargo, una clase puede ser un objeto.

El Enfoque Orientado a Objeto se basa en cuatro principios que constituyen la base de todo desarrollo orientado a objetos. Estos principios son: la Abstracción, el Encapsulamiento, la Modularidad y la Herencia.

Se ha considerado que las características son las siguientes:

- **Abstracción:** Denota las características esenciales de un objeto, donde se capturan sus comportamientos.
- **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.
- **Modularidad:** Se denomina Modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.
- **Principio de ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que específica cómo pueden interactuar con los objetos de la clase.
- **Polimorfismo:** Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.
- **Herencia:** Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación.
- **Recolección de basura:** La recolección de basura o garbage collector es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos.

Componentes

- Clase: Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- Herencia: (Por ejemplo, herencia de la clase C a la clase D) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D.
- Objeto: Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.
- Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un «mensaje». Desde el punto de vista del comportamiento, es lo que el objeto puede hacer.
- Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente.
- Mensaje: Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Propiedad o atributo: Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.
- Estado interno: Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos).
- Componentes de un objeto: Atributos, identidad, relaciones y métodos.
Identificación de un objeto: Un objeto se representa por medio de una tabla

o entidad que esté compuesta por sus atributos y funciones correspondientes.

ENFOQUES DE VANGUARDIA

Estos enfoques como su nombre lo dicen, son los nuevos paradigmas de la Ingeniería de Software que tratan de aportar una guía para aplicaciones informáticas que surgen a partir de nuevas tendencias tecnológicas y que requieren de un desarrollo en áreas cada vez más complejas y especializada. Los nuevos paradigmas de la ingeniería de software que se estudian por el momento son: La Programación Evolutiva, La Programación Ubicua y La Programación Orientada a Aspectos.

Programación evolutiva

Se encarga de simular el proceso evolutivo natural y llevarlo a la computación. La computación Evolutiva la conforman los Algoritmos Genéticos

Programación ubicua

Su objetivo es hacer que la tecnología se integre en los objetos, cosas, tareas y entornos cotidianos. Esta integración se debe realizar de forma que las personas se centren en las tareas que están haciendo, no en las herramientas que utilizan, porque se pretende que esas herramientas pasen desapercibidas. Está orientada más comúnmente hacia dispositivos móviles, para que sean "invisibles" a la perspectiva del usuario.

Programación orientada a aspectos

Desea darles a los profesionales un método que les permita manejar aplicaciones que son muy grandes, y que en el desarrollo puedan traer muchas complicaciones por su complejidad. Es por ello, que se está tratando de perfeccionar este paradigma.

CONCLUSIÓN

La Ingeniería de Software es reconocida como una disciplina legítima, digna de tener una investigación seria, un estudio cuidadoso y ha generado una gran controversia. En la industria el Ingeniero del software ha sustituido al programador como título de trabajo preferente. Los modelos de procesos de software, métodos de ingeniería de software y herramientas se han adoptado con éxito en el amplio espectro de las aplicaciones industriales. Los gestores y usuarios reconocen la necesidad de un enfoque más disciplinado del software.

El objetivo principal de la ingeniería en software es el convertir el desarrollo de software en un proceso formal mediante técnicas estructuradas a través de métodos de ingeniería, todo ello con resultados predecibles que permitan tener un producto final de alta calidad y satisfaciendo así las necesidades y expectativas del cliente el cual abarca la captura de requerimientos, diseño, desarrollo, pruebas y mantenimiento.

REFERENCIAS

- [1] "4.1enfoque Estructurado | PDF | Ingeniería de software | Software". Scribd. Accedido el 30 de diciembre de 2023. [En línea]. Disponible: <https://es.scribd.com/document/337591557/4-1enfoque-estructurado>

- [2] “Fundamentos del enfoque orientado a objetos”. Fundamentos del enfoque orientado a objetos. Accedido el 30 de diciembre de 2023. [En línea]. Disponible: <https://ingdelsoftwareseccion2.wordpress.com/>
- [3] “Ensayo "Paradigmas de La Ingeniería de Software" | PDF | Software | Ingeniería de software”. Scribd. Accedido el 30 de diciembre de 2023. [En línea]. Disponible: <https://es.scribd.com/document/541612996/Ensayo-Paradigmas-de-la-ingenieria-de-software>

VALOR : 40%

EVALUACIÓN UNIDAD 4. FUNDAMENTOS DE SISTEMAS DE INFORMACIÓN

INGENIERÍA INFORMATICA ITSSAT

Se ha registrado el correo del encuestado (221u0507@alumno.itssat.edu.mx) al enviar este formulario.

NOMBRE Y GRUPO *

Annette Montan Martínez, 310-A

GRUPO: *

310-A

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema. Un Caso de Uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Un Caso de Uso es una unidad de trabajo significativo; por ejemplo crear una solicitud y modificar una solicitud son todos Casos de Uso. * 5 puntos

 V F

Un actor es un usuario del sistema. Esto incluye usuarios humanos y otros sistemas computacionales. Un actor usa un Caso de Uso para ejecutar una porción de trabajo de valor para el negocio. El conjunto de casos de uso al que un actor tiene acceso define rol en el sistema y el alcance de su acción.

* 5 puntos

 V F

Los diagramas de secuencia muestran el flujo de mensajes de un objeto a otro y, como tales, representan los métodos y los eventos soportados por un/a objeto/clase.

* 5 puntos

 V F

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere).

* 5 puntos

 V F

Construcción: La construcción del sistema las fases dentro de esta etapa son las siguientes: Análisis: Se analiza el problema a resolver desde la perspectiva de los usuarios y de las entidades externas que van a solicitar servicios al sistema. Diseño: El sistema se especifica en detalle, describiendo cómo va a funcionar internamente para satisfacer lo especificado en el análisis. Implementación: Se lleva lo especificado en el diseño a un lenguaje de programación. Pruebas: Se llevan a cabo una serie de pruebas para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación y Especificación de Requisitos.

* 10 puntos

 V F

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

* 5 puntos

 V F

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre_del_objeto: nombre_de_la_clase

* 5 puntos

 V F

Google Formularios