



Universidad Tecnológica de Durango

Tecnologías de la Información

Programación Orientada a Objetos

Informe Técnico U1

“Sistema Informático en consola que permite administrar un
negocio de venta de ropa”

Alumnos:

- Bueno Cao Romero Erik Gabriel
- Guillén López José Alberto
- Jiménez Delgado Luis Héctor
- Lomas Corral Edson
- Meraz Sida Fernando

3°B BIS

Docente:

- Ing. Dagoberto Fiscal Gurrola, M.T.I.

Durango, Dgo.

Octubre 2025

Tabla de contenido

Tabla de contenido	2
Tabla de Ilustraciones	3
Objetivos	4
General	4
Específico	4
Actividades del reporte	5
Clases	5
Objetos	6
Casos de uso	7
Actividades	8
Secuencia	11
Estados	12
Conclusiones	13
Bibliografía	18

Tabla de Ilustraciones

Ilustración 1 Diagrama de clases.	5
Ilustración 2 Diagrama de objetos.	6
Ilustración 3 Diagrama de Casos de uso.	7
Ilustración 4 Diagrama de Actividades Pte.1.....	8
Ilustración 5 Diagrama de Actividades Pte.2.....	9
Ilustración 6 Diagrama de Actividades Pte.3.....	10
Ilustración 7 Diagrama de Secuencia.....	11
Ilustración 8 Diagrama de Estados.	12

Objetivos

General

- Analizará, diseñará y codificará algoritmos computacionales a través del paradigma de programación orientado a objetos para resolver problemas del ámbito productivo e implementar soluciones eficientes.

Específico

- Determinar los requisitos y requerimientos mediante el uso del lenguaje unificado de modelado UML para desarrollar una solución que satisfaga las necesidades específicas de una situación a resolver, así como representar clases en lenguaje de programación orientado a objetos para codificar el análisis y diseño de modelos en UML.

Actividades del reporte

Clases

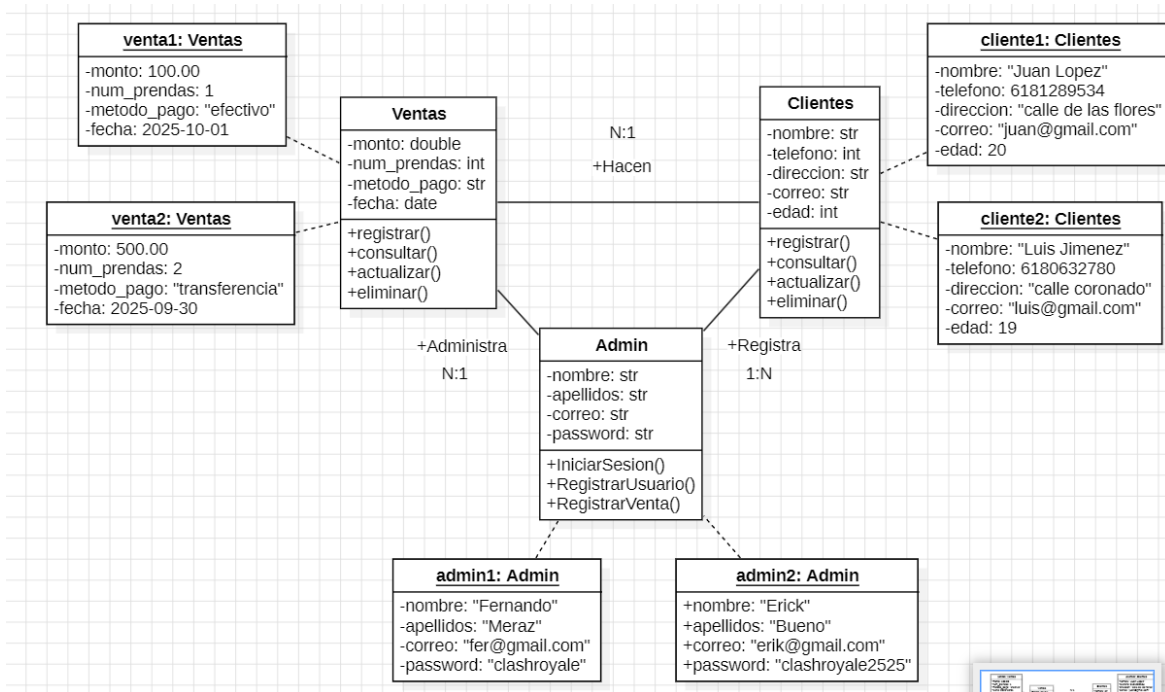


Ilustración 1 Diagrama de clases.

En la imagen anterior se muestra la evidencia de los diagramas de clases y objetos del proyecto integrador, en donde están creadas las clases Admin, Ventas y Clientes con atributos privados y métodos públicos y dos objetos por cada clase con los valores de sus atributos además de su asociación y cardinalidad.

Objetos

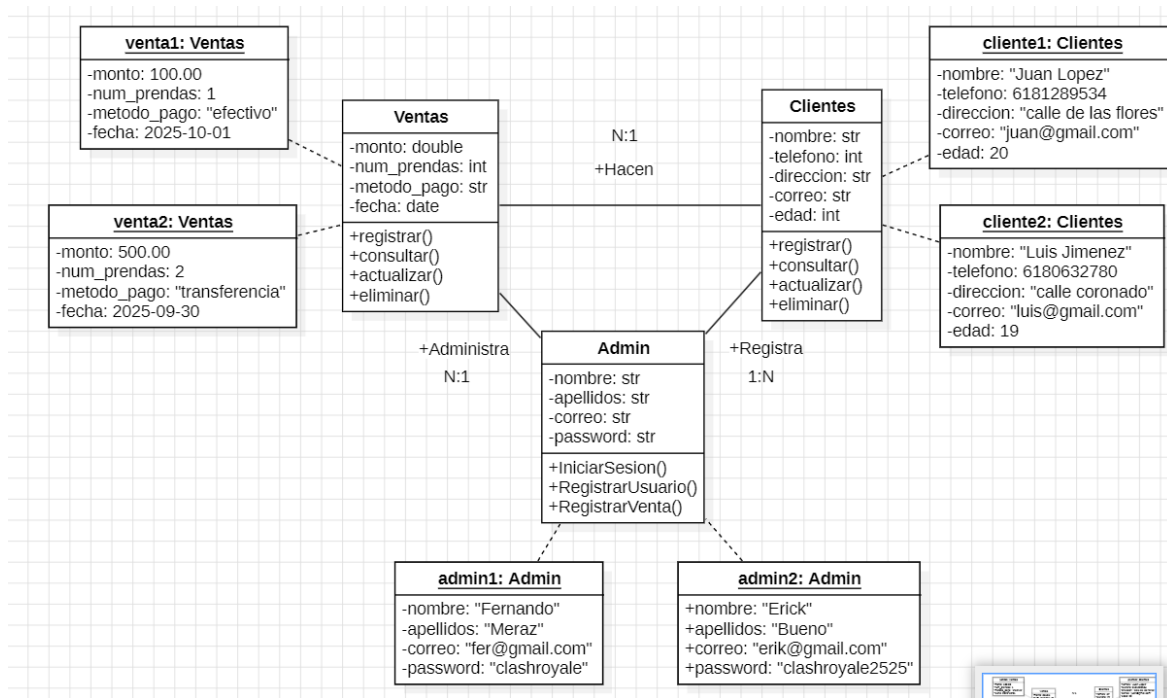


Ilustración 2 Diagrama de objetos.

En la imagen anterior se muestra la evidencia de los diagramas de clases y objetos del proyecto integrador, en donde están creadas las clases Admin, Ventas y Clientes con atributos privados y métodos públicos y dos objetos por cada clase con los valores de sus atributos además de su asociación y cardinalidad.

Casos de uso

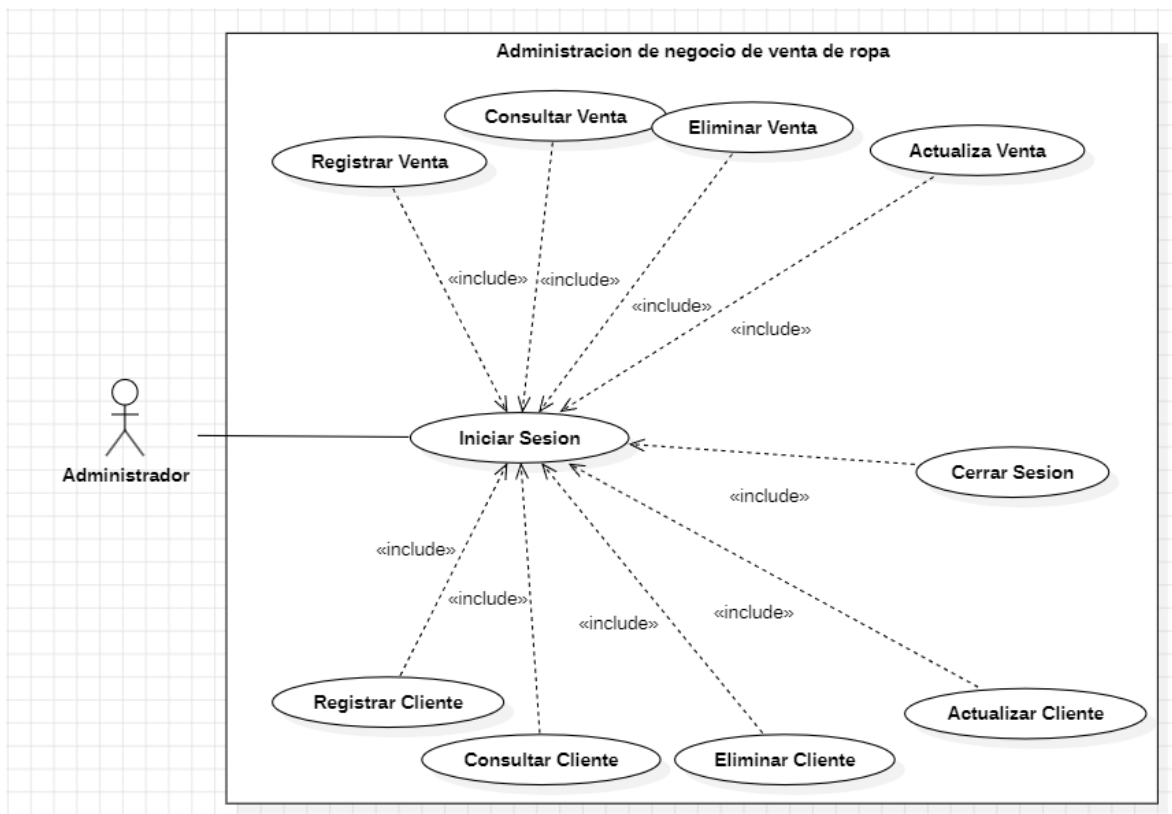


Ilustración 3 Diagrama de Casos de uso.

En la imagen anterior se muestra la evidencia del diagrama de casos de uso del proyecto integrador, donde se representa el actor Administrador, en donde su caso de uso principal es iniciar sesión y una vez que lo haga puede utilizar los demás para administrar clientes y ventas o cerrar sesión.

Actividades

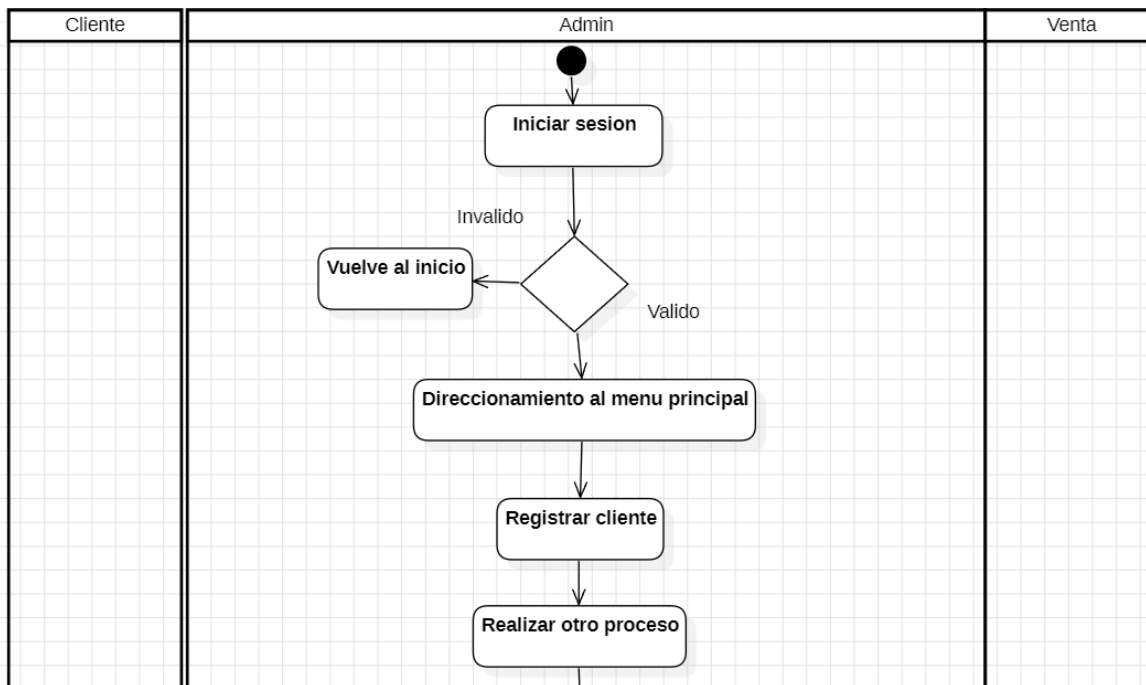


Ilustración 4 Diagrama de Actividades Pte.1

En la imagen anterior se muestra la evidencia de la primera parte del diagrama de actividades del proyecto integrador, en este se puede apreciar las diferentes actividades que realizan los objetos a lo largo del tiempo, contando con un inicio, las actividades, toma de decisiones y un fin.

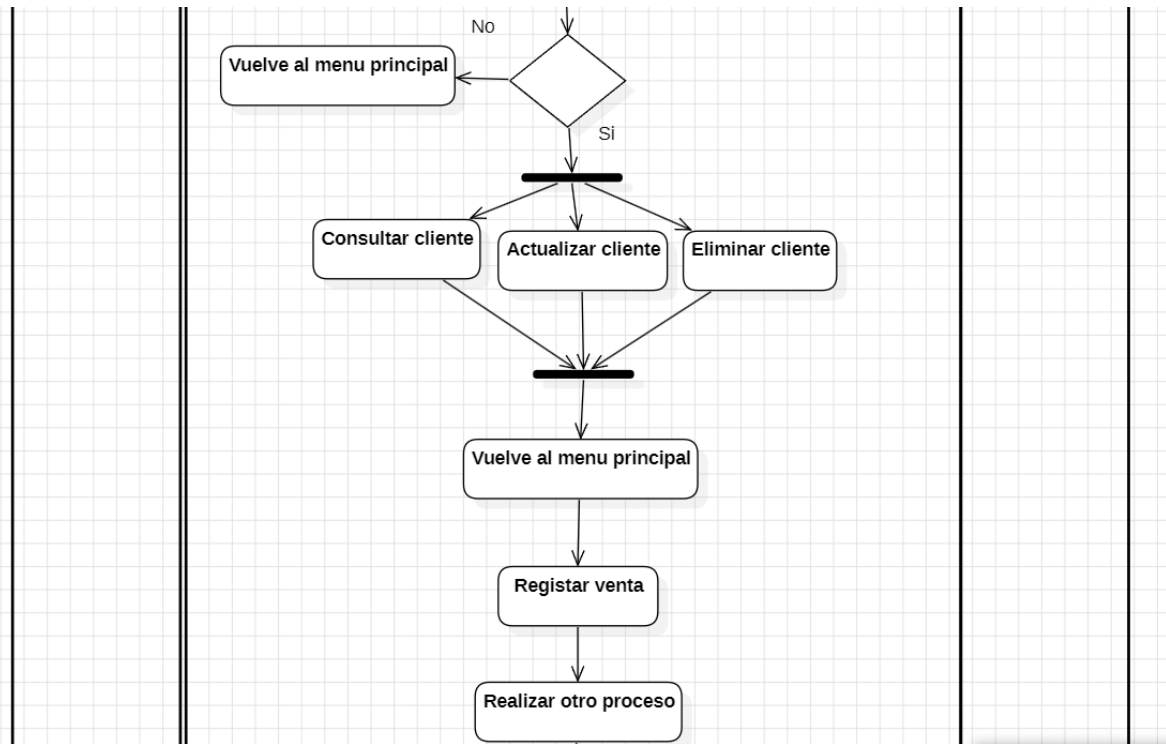


Ilustración 5 Diagrama de Actividades Pte.2

En la imagen anterior se muestra la evidencia de la segunda parte del diagrama de actividades del proyecto integrador, en este se puede apreciar las diferentes actividades que realizan los objetos a lo largo del tiempo, contando con un inicio, las actividades, toma de decisiones y un fin.

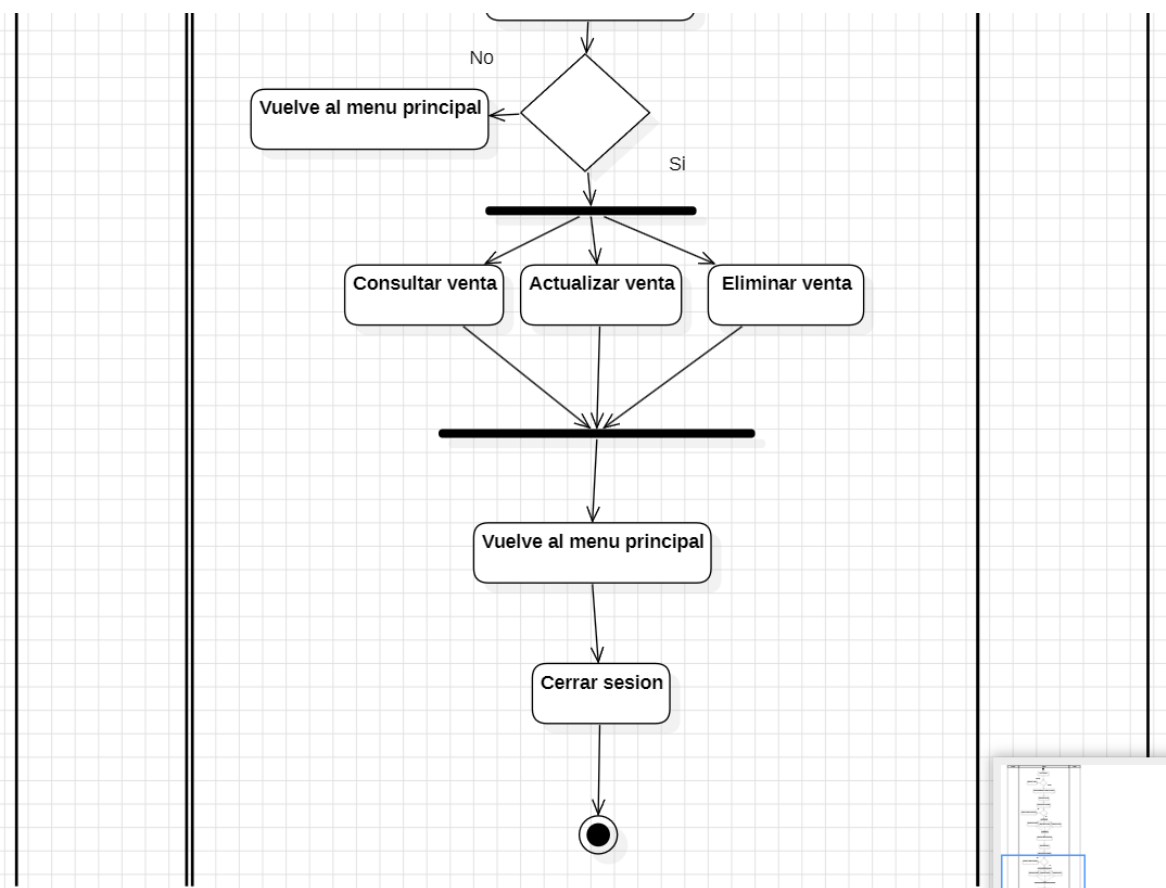


Ilustración 6 Diagrama de Actividades Pte.3

En la imagen anterior se muestra la evidencia de la tercera parte del diagrama de actividades del proyecto integrador, en este se puede apreciar las diferentes actividades que realizan los objetos a lo largo del tiempo, contando con un inicio, las actividades, toma de decisiones y un fin.

Secuencia

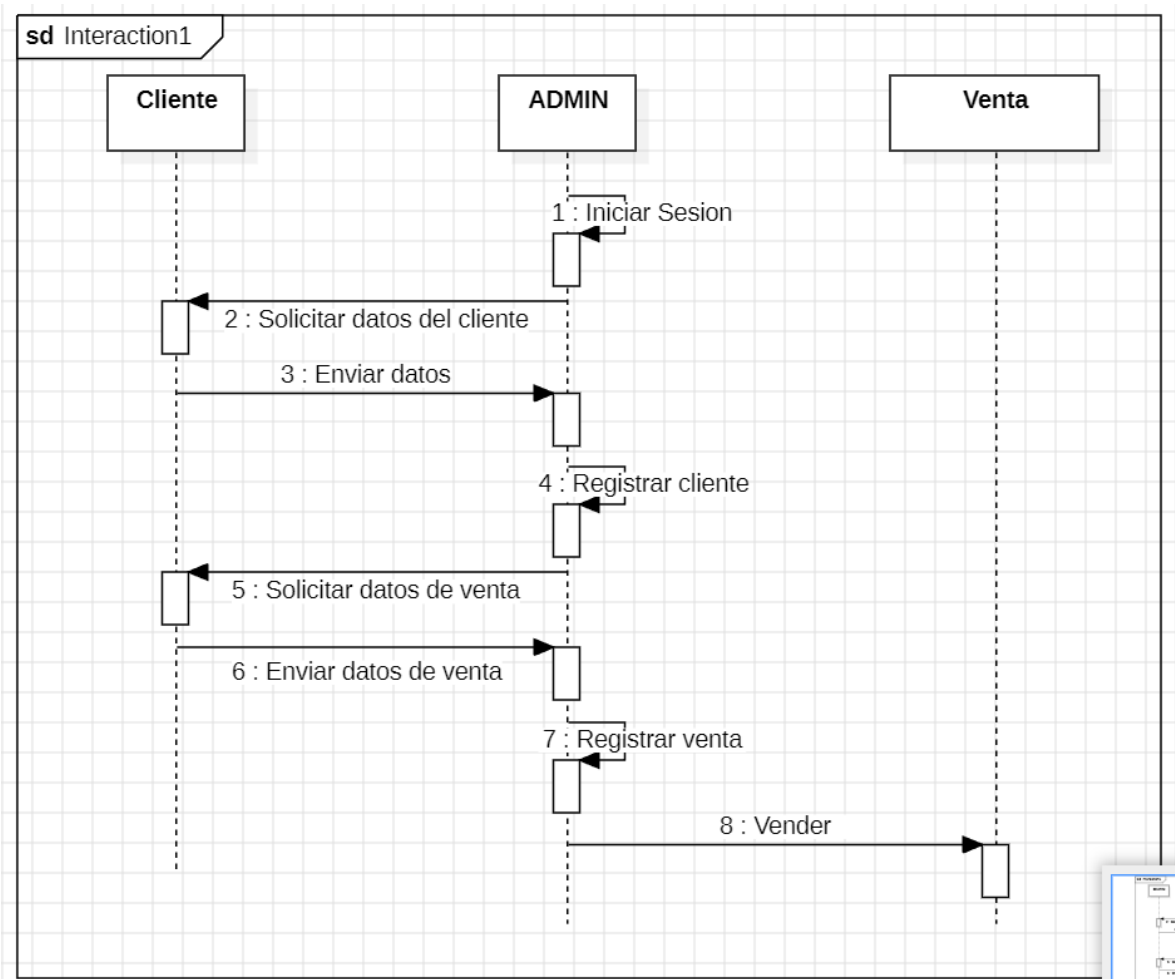


Ilustración 7 Diagrama de Secuencia.

En la imagen anterior se muestra la evidencia del diagrama de secuencia del proyecto integrador, en este se puede apreciar los diferentes procesos presentes en el sistema, también está presente la línea de vida de cada objeto para poder ver en qué parte específica de la ejecución se hace uso de ellos.

Estados

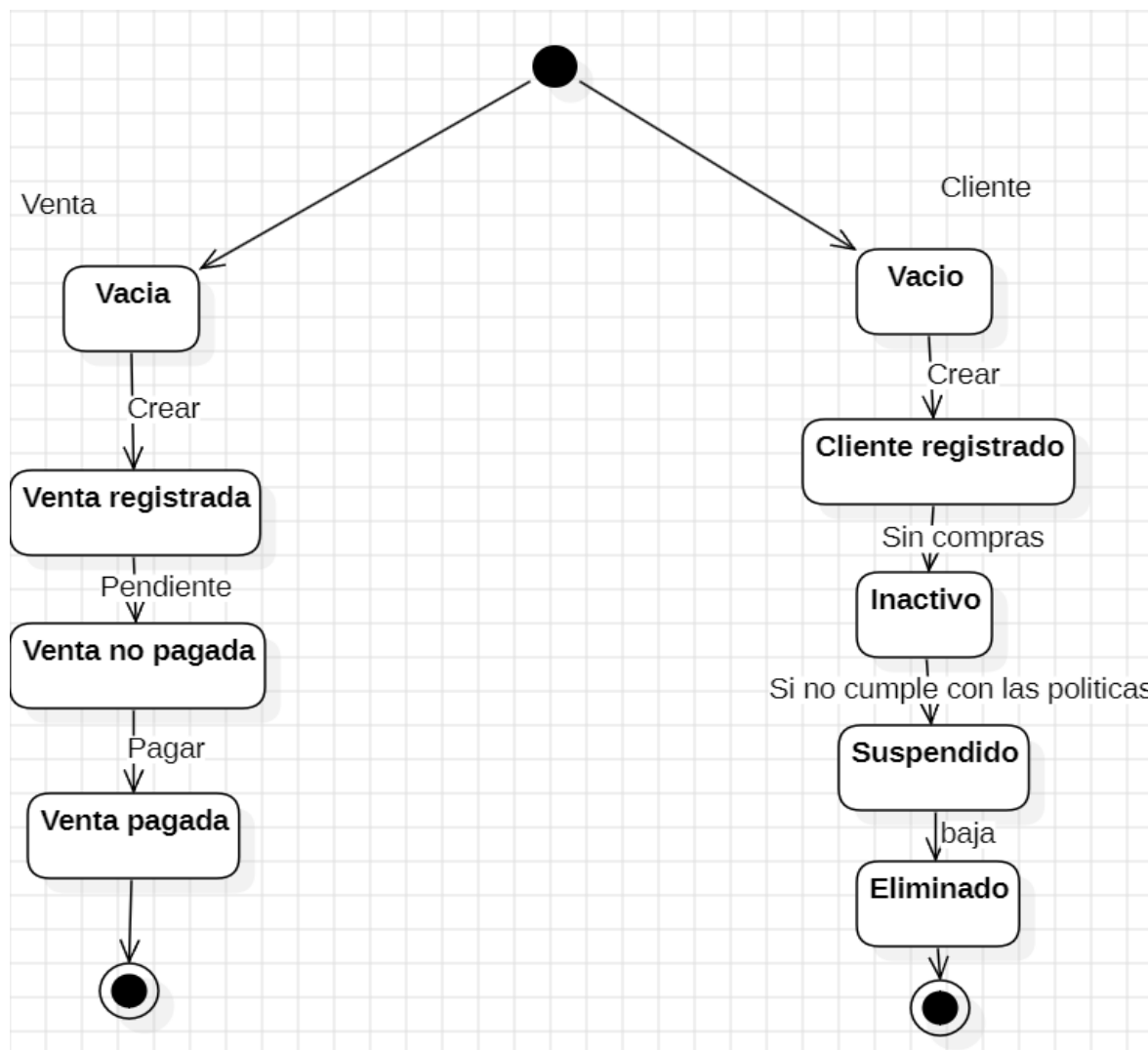


Ilustración 8 Diagrama de Estados.

En la imagen anterior se muestra la evidencia del diagrama de estados del proyecto integrador, en este se puede apreciar los diferentes estados por los que pasa cada objeto a lo largo de su ciclo de vida, en la izquierda se trata con los objetos de ventas y en la derecha con los de clientes.

Conclusiones

El desarrollo de este sistema de gestión de ropa permitió reforzar de manera práctica los conocimientos adquiridos a lo largo de la unidad número uno, donde el profesor nos introdujo al mundo de los diagramas UML, en los cuales, en total, existen catorce diferentes tipos de diagramas para visualizar diferentes partes de un sistema de software. En nuestro caso, se nos encargó realizar los primeros 6 tipos de diagramas UML, los cuales son: Clases, Objetos, Casos de Uso, Actividades, Secuencias y Estados.

A lo largo de la unidad se nos explicó de manera intensiva como nos ayudan estos diagramas para mejorar la creación, organización, manejo y administración de un sistema de software que se esta creando desde cero, al hacer esto, nos ayuda a que todas las personas de un equipo en el que se este trabajando entiendan y se relacionen de la mejor manera con el software para así poder estar todos en la misma página y trabajar de la mejor manera.

Al mismo tiempo, durante esta primera unidad del cuarto cuatrimestre de la carrera, procedimos a cambiar de paradigma de programación, del paradigma de programación estructurada, a un paradigma de programación orientada a objetos “POO”. Al hacer esto, dimos vuelta completamente a la página, cambiando nuestra forma de pensar y manejar código a la hora de la creación de un sistema de software, ayudándonos a encaminarnos y prepararnos mejor para nuestro futuro como ingenieros en Software.

Bueno Cao Romero Erik Gabriel

El desarrollo del sistema de gestión de ropa representó una práctica fundamental para consolidar lo aprendido en la primera unidad. El enfoque en los diagramas UML nos permitió descubrir cómo un mismo sistema puede visualizarse desde distintas perspectivas, lo cual facilita tanto su análisis como su construcción. Realizar los seis diagramas principales Clases, Objetos, Casos de Uso, Actividades, Secuencias y Estados, fue una experiencia enriquecedora, ya que cada uno aportó un nivel de detalle específico sobre la estructura y el funcionamiento del software. Esto no solo reforzó nuestros conocimientos teóricos, sino que nos dio una herramienta muy útil para aplicarla en futuros proyectos.

Al mismo tiempo, este trabajo nos ayudó a comprender que los diagramas no son únicamente un requisito académico, sino una forma de comunicación efectiva dentro de un equipo. Gracias a ellos, todos los integrantes logramos ponernos en sintonía y entender mejor la lógica detrás del sistema que se estaba construyendo. Esta práctica resultó clave para mejorar el trabajo colaborativo y para darnos cuenta de que la planificación gráfica es tan importante como la implementación del código en sí misma.

Por último, la transición del paradigma de programación estructurada al paradigma orientado a objetos marcó una diferencia notable en nuestro proceso de aprendizaje. Este cambio nos obligó a replantear la manera de organizar nuestras ideas y de pensar en términos de clases y objetos en lugar de procedimientos lineales. Aunque al inicio supuso un reto, poco a poco se convirtió en una herramienta poderosa para diseñar software más modular, reutilizable y escalable. Sin duda, esta unidad nos preparó de una mejor manera para enfrentar los retos de los siguientes cuatrimestres y para nuestro futuro profesional como ingenieros en software.

Guillén López José Alberto

El trabajo realizado en el sistema de gestión de ropa fue una experiencia clave para afianzar los conceptos aprendidos en la unidad uno. El estudio y aplicación de los diagramas UML fue una de las partes más significativas, ya que nos permitió comprender de manera más detallada cómo se puede representar el diseño de un sistema antes de su implementación. Cada diagrama aportó un nivel distinto de análisis: desde las clases y objetos hasta los estados y actividades, lo que nos dio una visión más completa del proceso de desarrollo de software.

Uno de los aprendizajes más valiosos fue descubrir que los diagramas son una herramienta que va más allá de lo técnico, pues también cumplen una función comunicativa dentro del equipo. Gracias a ellos logramos mantener un lenguaje común y facilitar la comprensión del proyecto, evitando confusiones y trabajando de manera más organizada. Este aspecto nos enseñó que el desarrollo de software no se trata solo de programar, sino también de planear, comunicar y trabajar en conjunto para alcanzar un objetivo en común.

Además, el cambio de paradigma hacia la programación orientada a objetos nos permitió evolucionar nuestra manera de pensar. Pasamos de resolver problemas con estructuras secuenciales a diseñar soluciones basadas en objetos, con mayor modularidad y claridad. Este cambio fue fundamental para abrirnos la mente hacia un estilo de programación más cercano a la realidad profesional. Sin duda, este primer acercamiento nos ha dado bases sólidas para enfrentar proyectos de mayor complejidad y para continuar avanzando en nuestra carrera como futuros ingenieros en software.

Jiménez Delgado Luis Héctor

La creación del sistema de gestión de ropa representó una oportunidad importante para aplicar los conocimientos adquiridos en la primera unidad. El aprendizaje de los diagramas UML fue esencial, ya que a través de los seis primeros logramos entender cómo se estructura, organiza y comunica el diseño de un sistema de software. Esta experiencia nos mostró que planear y representar gráficamente las partes de un sistema es fundamental antes de proceder con su implementación, pues permite detectar errores y mejorar la claridad del proyecto.

Al mismo tiempo, este trabajo nos ayudó a darnos cuenta de la importancia del trabajo en equipo. Al elaborar los diagramas, fue posible que todos los integrantes compartiéramos la misma visión del proyecto, lo que facilitó la organización y la coordinación en cada etapa. De esta manera, aprendimos que los diagramas UML son una herramienta indispensable no solo para el análisis, sino también para fortalecer la comunicación entre desarrolladores.

Finalmente, la transición de la programación estructurada a la programación orientada a objetos fue uno de los retos más significativos de la unidad. Este cambio nos obligó a adoptar un nuevo enfoque en la forma de diseñar y programar, lo que en un principio representó una dificultad, pero con la práctica se convirtió en una ventaja. Comprender los conceptos de clases, objetos, métodos y atributos nos permitió organizarnos mejor y adquirir una perspectiva más profesional del desarrollo de software. En conclusión, este proyecto fue clave para consolidar conocimientos, mejorar el trabajo colaborativo y prepararnos para futuros desafíos académicos y laborales.

Lomas Corral Edson

El proyecto del sistema de gestión de ropa fue una experiencia académica que nos permitió consolidar los aprendizajes de la primera unidad de una manera práctica. Al trabajar con los diagramas UML logramos comprender cómo se puede descomponer un sistema en diferentes representaciones que facilitan su análisis y diseño. Realizar los diagramas de Clases, Objetos, Casos de Uso, Actividades, Secuencias y Estados nos dio una visión más amplia sobre el funcionamiento interno y externo de un software, lo que resultó fundamental para conectar la teoría vista en clase con la práctica de desarrollo.

Asimismo, esta actividad nos mostró la importancia que tiene la organización en el desarrollo de un proyecto de software. Gracias al uso de los diagramas, todos los integrantes del equipo pudimos compartir la misma perspectiva del sistema y trabajar de manera coordinada. Esto nos enseñó que un buen diseño no solo facilita la implementación del código, sino que también mejora la comunicación y la colaboración entre las personas que participan en el proyecto. En ese sentido, este ejercicio nos ayudó a valorar la utilidad de los UML como un lenguaje común entre programadores y diseñadores.

Por último, el cambio hacia el paradigma de programación orientada a objetos fue un paso decisivo en nuestra formación. Este nuevo enfoque nos llevó a modificar la manera en que pensamos y organizamos nuestras soluciones, pasando de estructuras rígidas y lineales a modelos más flexibles y reutilizables. Aunque al inicio representó un reto, este cambio resultó enriquecedor y nos acercó más a las metodologías que se utilizan en el mundo profesional. Sin duda, este proyecto marcó un avance importante en nuestro aprendizaje y nos preparó mejor para los siguientes retos académicos y profesionales.

Meraz Sida Fernando

Bibliografía

- El libro de Python. (s.f.). Programación orientada a objetos en Python. Recuperado de: <https://ellibrodepython.com/programacion-orientada-a-objetos-python>
- Miro. (s.f.). ¿Qué es un diagrama UML? Recuperado de: <https://miro.com/es/diagrama/que-es-diagrama-uml/#diagramas-estructurales>
- Lucidchart. (s.f.). Tutorial de diagrama de clases UML. Recuperado de <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>