

**EVIDENCIA GA2-220501093-AA2-EVO1: ELABORACIÓN DE LOS
DIAGRAMAS DEL MODELO DE DOMINIO DE PROYECTO.**

**RICARDO ANDRES CORONEL MERCADO
OLIVER NAY DOMINGUEZ LEON**

ANALISIS Y DESARROLLO DE SOFTWARE

SENA

FICHA: 28344869

INSTRUCTORA AREA TECNICA: SANDRA LILIANA HUERTAS

CONTENIDO

- 1- ¿CUAL ES LA IMPORATANCIA DE UML?
- 2- ¿CUALES SON LOS DIAGRAMAS QUE PUEDEN REPRESENTARSE CON UML?
- 3- ¿QUE HERRAMIENTAS DE SOFTWARE EXISTEN EN EL MERCADO PARA UML?
- 4- DIAGRAMA DE CASO DE USOS (USUARIOS Y FUNCIONES)
- 5- DIAGRAMA DE ACTIVIDADES
- 6- DIAGRAMA DE CLASES

La importancia de UML a la hora de desarrollar software se basa en las siguientes razones:

- Una Comunicación muy clara UML es un lenguaje estándar y visual para comunicar diseños de software entre miembros del equipo, clientes y otras partes interesadas. Esto nos reduce la ambigüedad y facilita la comprensión de los conceptos de diseño.
- Excelente documentación UML es un lenguaje que nos permite documentar de manera efectiva todos los aspectos del diseño de software, desde la estructura hasta el comportamiento, lo que nos ayuda a mantener un registro claro de las decisiones de diseño y facilita la comprensión del sistema para futuros desarrollos y mantenimiento.
- Un muy buen diseño antes de la implementación es muy importante UML permite a los desarrolladores visualizar y planificar el diseño del software antes de escribir código. Esto nos ayuda a identificar posibles problemas de diseño, mejorar la estructura del sistema y optimizar su arquitectura antes de invertir tiempo y recursos en la implementación.
- Manejo óptimo de Reutilización de componentes mediante el modelado de componentes y sus relaciones en UML, nosotros los desarrolladores podemos identificar oportunidades para la reutilización de código y diseñar sistemas modulares que sean más fáciles de mantener y extender.
- Capacidad de análisis y diseño orientados a objetos UML está diseñado para modelar sistemas orientados a objetos, lo que nos facilita el análisis y diseño de sistemas complejos al proporcionar herramientas y notaciones específicas para representar clases, objetos, herencia, asociaciones y otros conceptos fundamentales de la programación orientada a objetos.
- En resumen, podemos concluir que UML es muy importante porque nos ayuda a los equipos de desarrollo de software a comunicarnos de manera efectiva, documentar nuestros diseños, planificar y visualizar sistemas antes de que los implementemos, y facilitar el análisis y diseño orientados a objetos.

LOS DIAGRAMAS QUE PODEMOS REPRESENTAR CON EL LENGUAJE UML SON LOS SIGUIENTES:

Diagrama de Casos de Uso: Mediante este podemos ver las interacciones entre un sistema y sus actores externos, mostrando los diferentes casos de uso y cómo los actores interactúan con el sistema.

Diagrama de Clases: Nos describe la estructura estática del sistema modelando las clases, sus atributos, métodos y relaciones entre ellas.

Diagrama de Objetos: Es similar al diagrama de clases, pero se enfoca en una instancia particular del sistema en un momento específico, mostrando objetos y sus relaciones.

Diagrama de Secuencia: Permite modelar la interacción entre objetos en secuencia temporal, mostrando los mensajes intercambiados entre ellos en un escenario particular.

Diagrama de Colaboración: Es muy similar al diagrama de secuencia, pero muestra la estructura estática de los objetos y sus relaciones, además de la interacción entre ellos.

Diagrama de Estados: hace una representación del comportamiento de un objeto a lo largo del tiempo, mostrando los diferentes estados en los que puede estar y las transiciones entre ellos.

Diagrama de Actividades: No muestra explícitamente el flujo de control de un proceso o algoritmo, mostrando las actividades y las decisiones que se toman en el camino.

Diagrama de Componentes: Mediante este podemos representar los componentes físicos del sistema y sus relaciones, mostrando cómo se agrupan en paquetes y cómo se comunican entre sí.

Diagrama de Despliegue: Define y describe los componentes físicos del sistema de cómo se distribuyen en el entorno de ejecución, mostrando nodos y sus relaciones de despliegue.

Diagrama de Paquetes: Es capaz de ordenar los elementos del modelo en grupos lógicos, mostrando la estructura de alto nivel del sistema y las dependencias entre los paquetes.

Algunas de las herramientas de software que pudimos identificar para nuestro lenguaje UML son:

Astah: Anteriormente conocido como Jude, Astah es una herramienta de modelado UML muy popular que ofrece diferentes ediciones según las necesidades del usuario, desde la edición profesional hasta la edición comunitaria gratuita.

IBM Rational Software Architect: Esta herramienta proporciona un entorno integrado para el diseño y desarrollo de software, incluido el modelado UML.

UMLet: Es una herramienta de diagramación UML de código abierto que ofrece una interfaz sencilla y rápida para crear diagramas UML.

Lucidchart: Aunque no está específicamente diseñado solo para UML, Lucidchart es una herramienta de diagramación en línea que ofrece plantillas y herramientas para crear diagramas UML de manera fácil y colaborativa.

Enterprise Architect: Es una herramienta completa para modelado UML que ofrece una amplia gama de características para el modelado de sistemas y procesos empresariales.

Visual Paradigm: Es una suite de modelado de software que incluye herramientas para UML, BPMN, arquitectura empresarial y más. Ofrece una interfaz intuitiva y opciones de colaboración en equipo.

TIPOS DE DIAGRAMAS

CASO DE USO

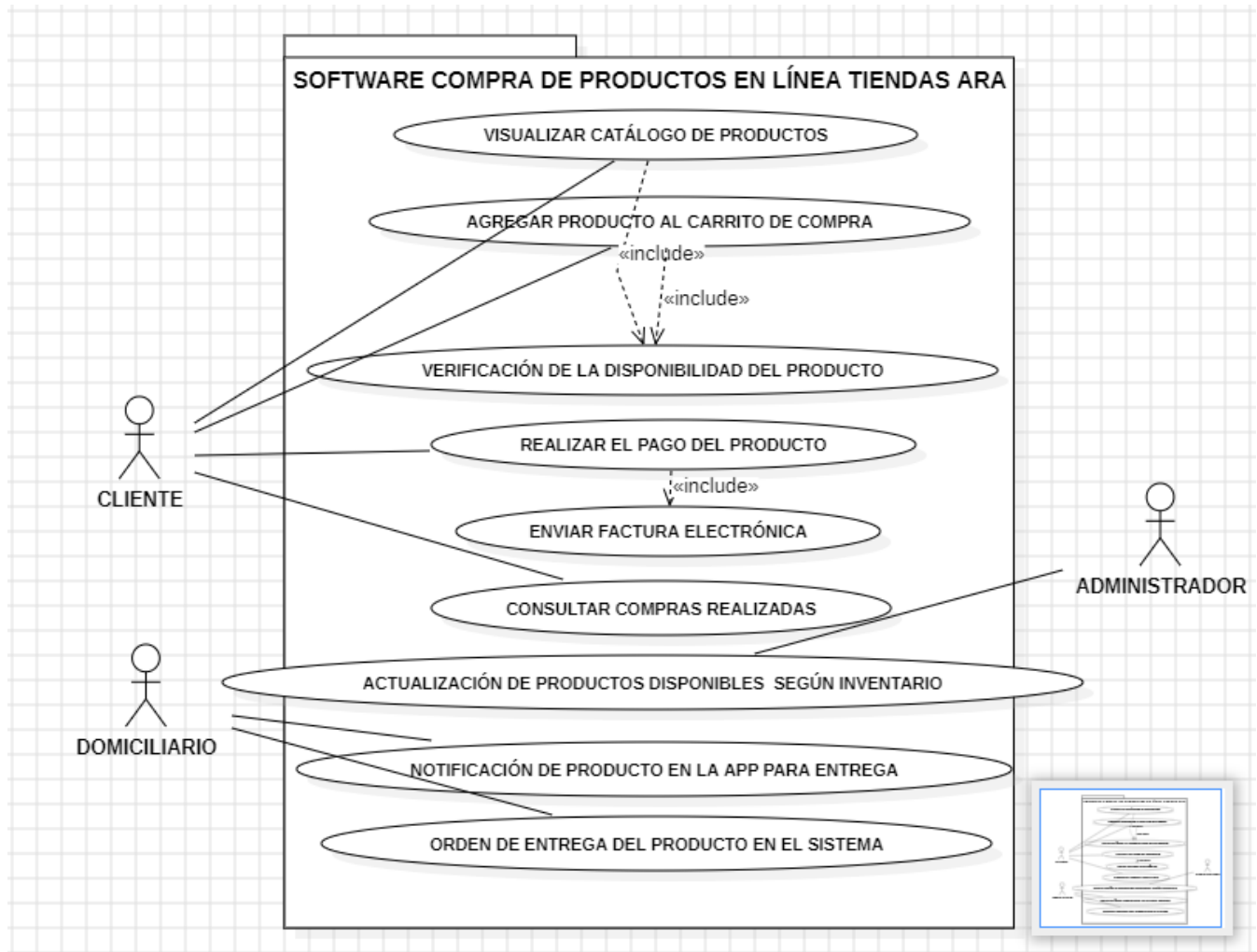


DIAGRAMA DE ACTIVIDADES

★ DIAGRAMA DE ACTIVIDADES TU ARA.mdj — StarUML (UNREGISTERED)
File Edit Format Model Tools View Window Debug Help

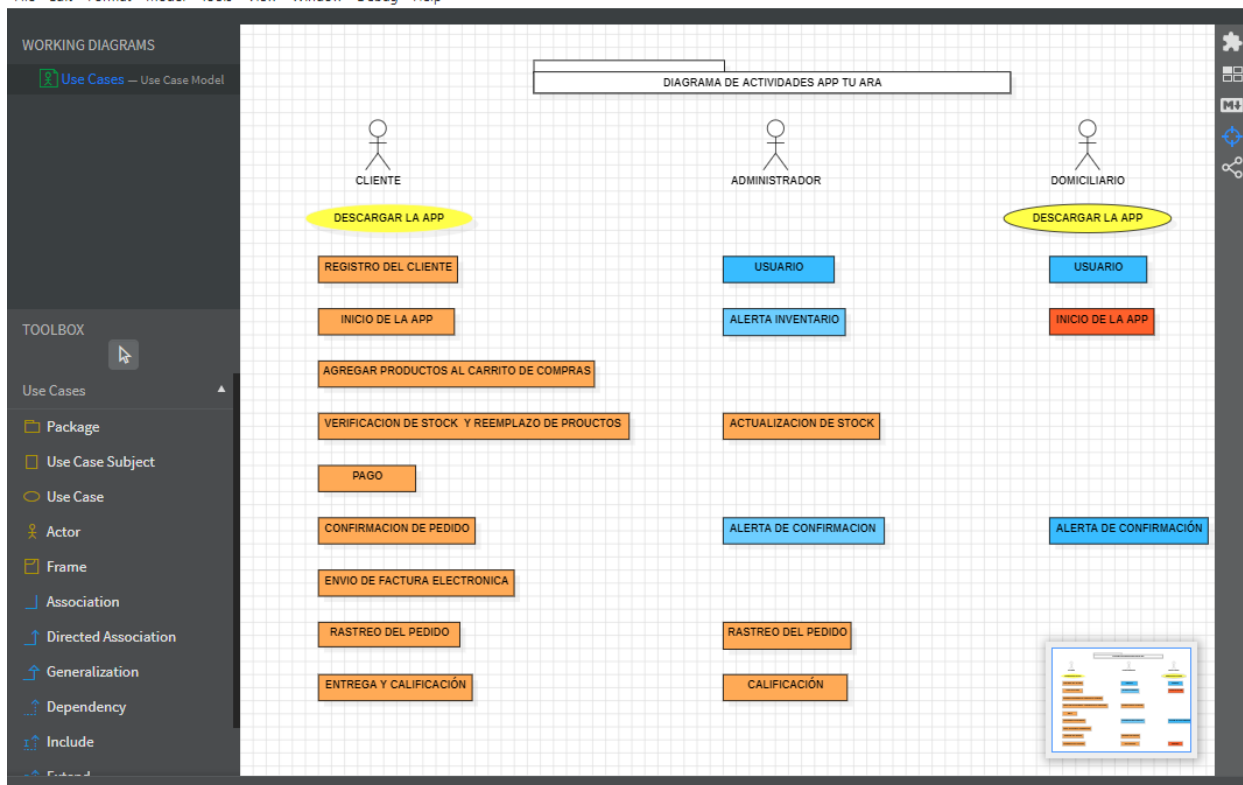


DIAGRAMA DE CLASES

