



UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Práctica 9 - Diseño lógico-físico

Nombre: Oscar Evanilson Gutiérrez Pérez, Código: 219748308

Nombre: Milton Osvaldo Rodriguez González, Código: 216572357

Materia: Seminario de Solución de Problemas de Ingeniería de Software

Sección: D01

Carrera: Ingeniería en Computación

7 de noviembre de 2021

Índice

Introducción	3
Desarrollo y Resultados	4
Conclusiones	9
Referencias	9

Introducción

En la actividad de esta semana vamos a presentar la última parte de la documentación de nuestro proyecto, en este caso hablamos del diseño lógico-físico de nuestro programa. Este es el último paso anterior a la construcción y desarrollo del software y es una parte importante de la planificación necesaria para la entrega del mismo.

Para la presentación del diseño lógico-físico de nuestro proyecto realizaremos los siguientes diagramas: Diagrama de estado, Diagrama de actividades, Diagrama de componentes y Diagrama de despliegue.

Los diagramas de estado son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede entrar un objeto particular y la manera en que cambia el estado del objeto, como resultado de los eventos que llegan a él. En la mayor parte de las técnicas Orientadas a Objetos, los diagramas de estado se dibujan para una sola clase, mostrando el comportamiento de un solo objeto durante todo su ciclo de vida (*diagramas de estado* - *eduardoaguirre*, s. f.).

El diagrama de actividades es un diagrama UML de comportamiento que muestra el flujo de control o el flujo de objetos, con especial énfasis en la secuencia y las condiciones de este flujo. Los diagramas de actividades muestran una secuencia de acciones, un flujo de trabajo que va desde un punto inicial hasta un punto final («▷ *Diagrama de actividades*», s. f.).

Los diagramas de componentes UML representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

En el contexto del UML, los componentes son partes modulares de un sistema independiente entre sí, que pueden reemplazarse con componentes equivalentes. Son autocontenidos y encapsulan estructuras de cualquier grado de complejidad. Los elementos encapsulados solo se comunican con los otros a

través de interfaces. Los componentes no solo pueden proporcionar sus propias interfaces, sino que también pueden utilizar las interfaces de otros componentes, por ejemplo, para acceder a sus funciones y servicios. A su vez, las interfaces de un diagrama de componentes documentan las relaciones y dependencias en una arquitectura de software (*Diagrama de componentes*, s. f.).

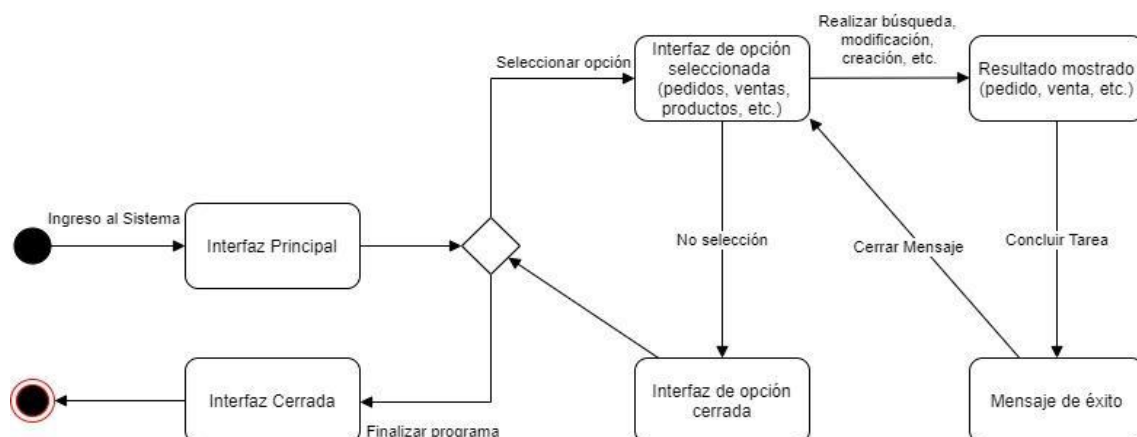
Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta. Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware

Los diagramas de despliegue ayudan a modelar la topología de hardware de un sistema en comparación con otros tipos de diagramas UML, que en su mayoría esbozan los componentes lógicos de un sistema (*Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue?*, s. f.).

Sabiendo todo lo mencionado anteriormente, damos paso a la presentación de estos diagramas que realizamos en base a el proyecto en desarrollo del programa administrador para la empresa “Tabacos Ornelas”.

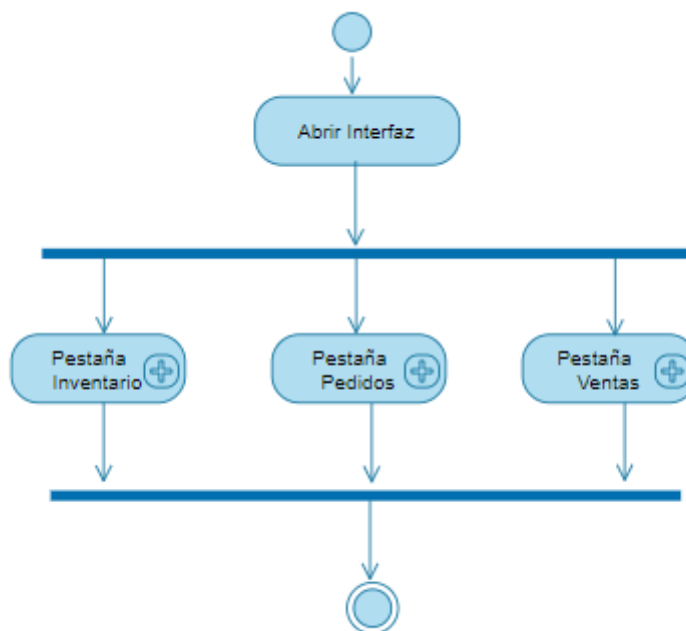
Desarrollo y Resultados

Diagrama de Estado:

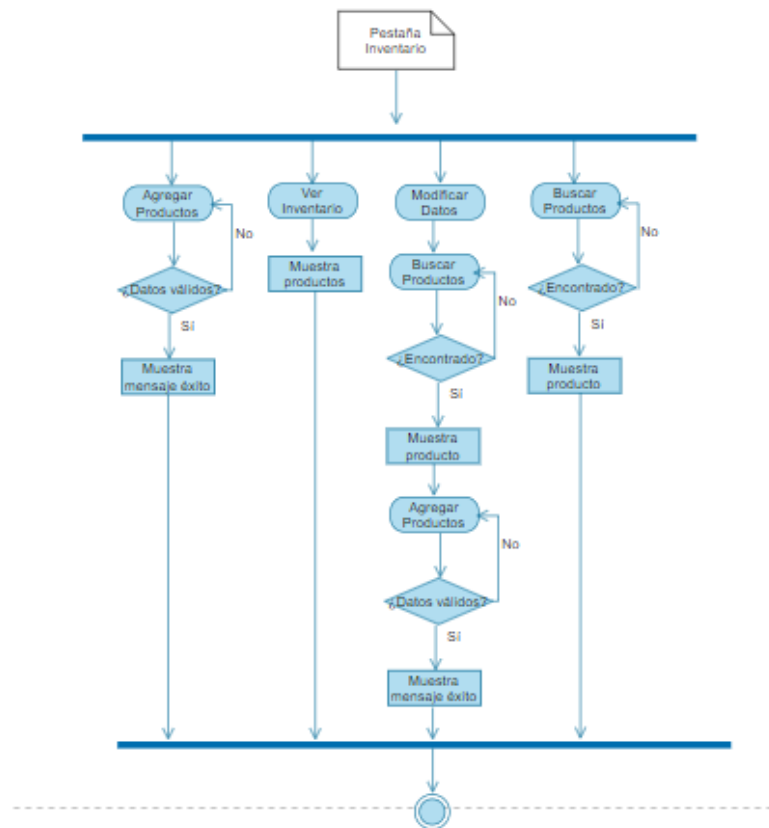


En este diagrama podemos observar los estados que se pueden presentar en nuestro sistema, por cada interfaz o tarea tenemos un estado, que dependiendo de las acciones que el usuario realice o las opciones que seleccione, irá realizando transiciones a las interfaces seleccionadas. Un ejemplo sería ingresar al sistema y escoger la interfaz de pedido, ingresamos a la interfaz de los pedidos y realizamos una búsqueda, nos muestra el resultado en pantalla y nos regresa a la interfaz principal, donde podemos seleccionar otra opción o salir del sistema.

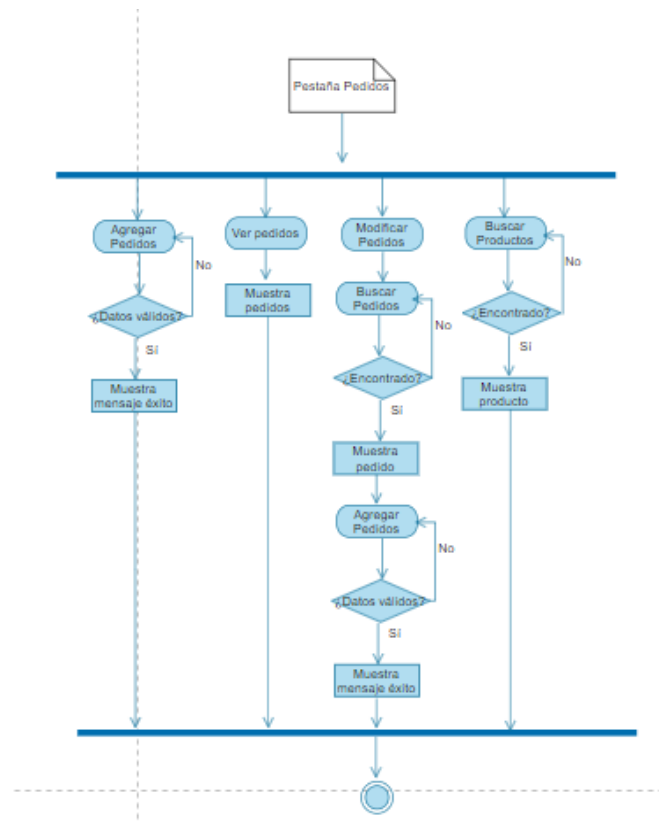
Diagrama de Actividades:



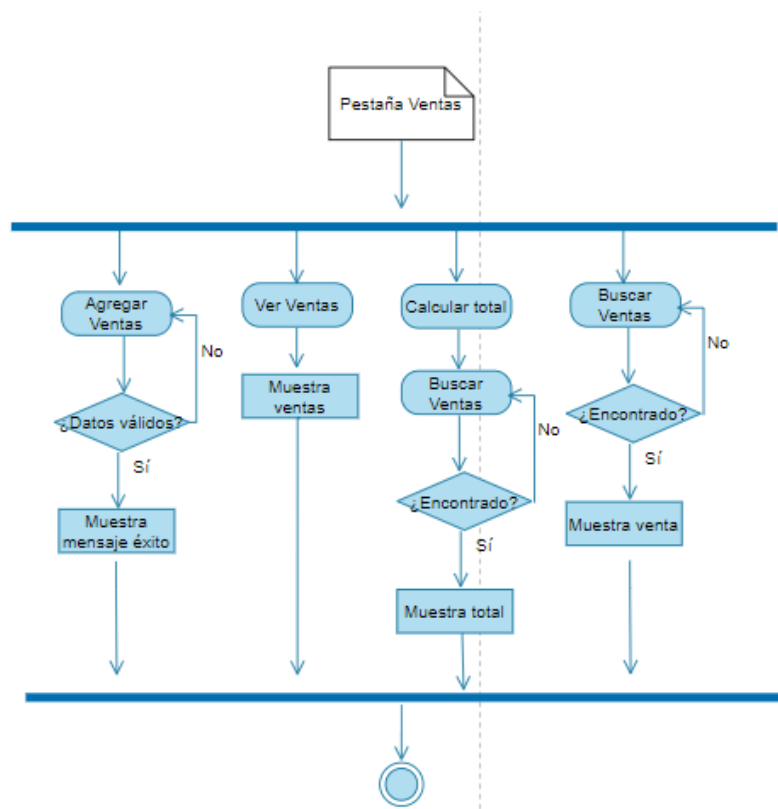
En el siguiente diagrama de actividades se puede observar que incluí tres subprocesos. Esto fue porque el diagrama se había hecho demasiado grande, por lo que decidí partirlo en cuatro partes.



Este es el primer subprocesso, muestra las opciones disponibles en la pestaña de inventario. Me basé en el diagrama de secuencias para construir este diagrama.

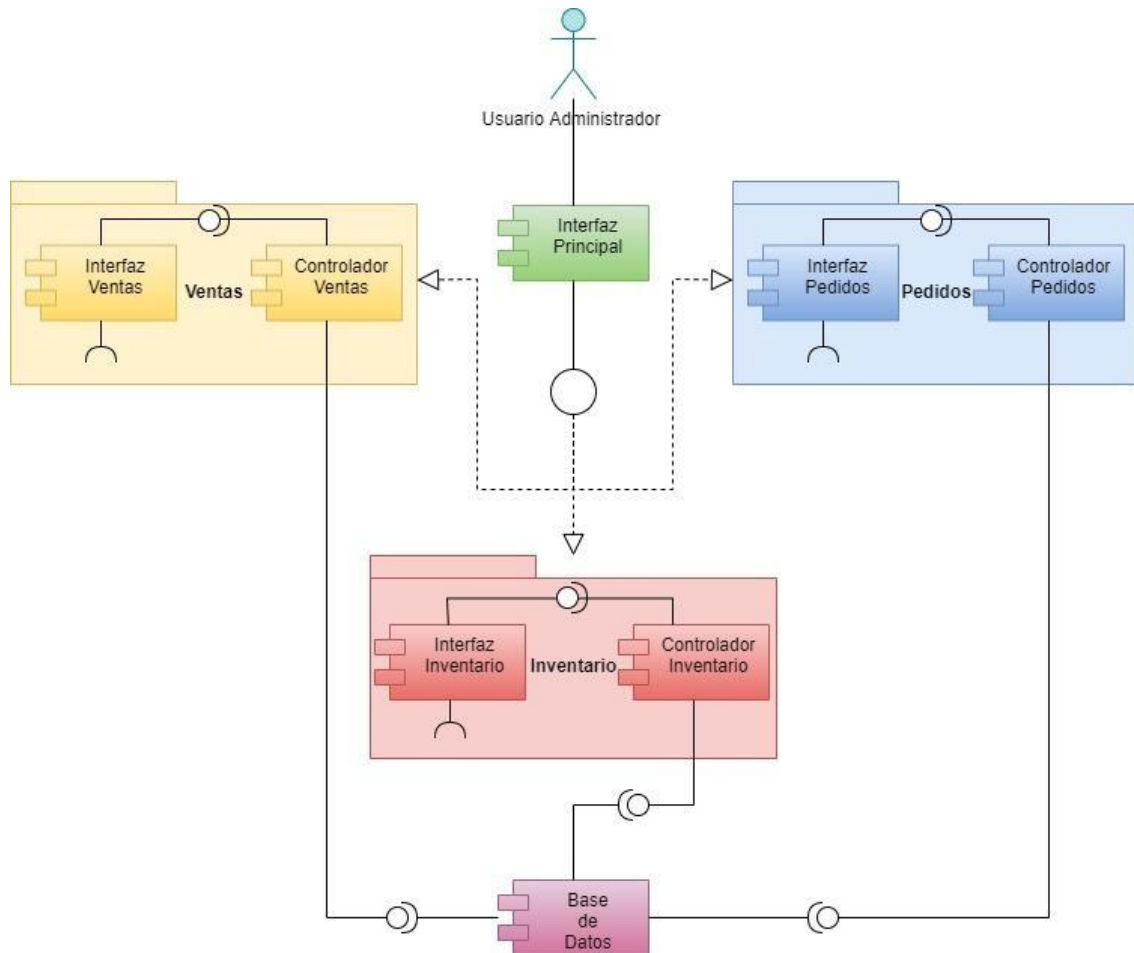


El segundo subproceso es el de las opciones existentes en la pestaña de pedidos. Es igual a la de la pestaña de inventario.



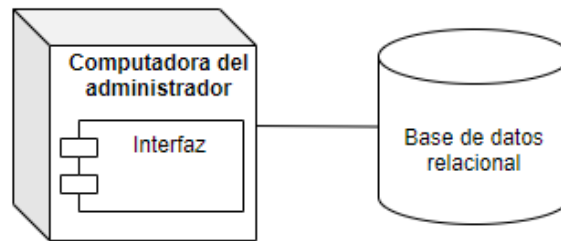
El último subproceso es el de ventas. Es un poco más pequeño que el resto, pues no incluye la opción de modificar, en su lugar tiene una función de calcular total (total de ventas en cierta fecha).

Diagrama de Componentes:



En este diagrama podemos ver los componentes de nuestro sistema y las relaciones que llevan entre estos, encontramos que por cada clase (inventario, ventas y pedidos) hay una interfaz y un controlador que nos ayuda a completar las tareas de cada clase, y estos tres controladores están conectados a una base de datos donde se almacenará la información de cada clase. El usuario puede acceder a estas interfaces desde una interfaz principal.

Diagrama de Despliegue:



El diagrama de despliegue que presentamos es bastante simple, incluso pensamos en no incluirlo por su simplicidad. Debido a que nuestro sistema y la base de datos se encuentran dentro de la misma computadora del cliente o administrador, nuestro diagrama quedó sumamente pequeño.

Conclusiones

Realizar esta actividad fue de mucha utilidad para mí, ya que por ejemplo mientras estaba haciendo el diagrama de actividades, me estuve apoyando de diagramas que realicé anteriormente, cosa que me hizo darme cuenta de la importancia de los diagramas realizados en el presente proyecto, también descubrí un error en el planteamiento de nuestro sistema, afortunadamente los diagramas sirvieron para encontrarlo antes de empezar a codificar, por lo cual lo vamos a agregar esos pequeños cambios en el sistema al momento de crearlo en código.

Algo que no me quedó muy claro fue el diagrama de despliegue, pues en nuestro caso era demasiado simple (o al menos eso creo yo), por lo cual no pude poner en uso todos los elementos de este.

Referencias

1. *diagramas de estado*—Eduardoaguirre. (s. f.). Recuperado 3 de noviembre de 2021, de <https://sites.google.com/site/eduardoaguirrevillalba/diagramas-de>
2. ▷ Diagrama de actividades. (s. f.). *DiagramasUML.com*. Recuperado 3 de noviembre de 2021, de <https://diagramasuml.com/actividades/>

3. *Diagrama de componentes: Modelado eficiente de sistemas con módulos de software.* (s. f.). IONOS Digitalguide. Recuperado 3 de noviembre de 2021, de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>
4. *Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue?*. (s. f.). Recuperado 3 de noviembre de 2021, de <https://createy.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>