

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

"DESARROLLO DE UN SISTEMA PARA GESTIONAR ÓRDENES DE UN RESTAURANTE, USANDO MVC"

Nombre:

Roberto Carlos Gallardo Andrade

Director: Ing. Fabián de la Cruz

Ciudad Quito, Año 2018

Dedicatoria

Este presente proyecto está dedicado principalmente a mi amada familia, por ser un gran apoyo cuando busco completar una meta y apoyarme siempre todo el tiempo, impulsándome a ser una mejor persona.

A mis amigos y compañeros que estuvieron en todo momento para extenderme una mano, en especial en momentos difíciles a lo largo de la carrera.

A Dios por poner en mi camino a todos los mencionados.

Agradecimiento

Quiero agradecer al restaurante CHOCHOTE por darme la oportunidad de realizar mi disertación en su empresa ayudándome con requerimientos y pruebas del sistema.

A mi director de tesis el ing. Fabián de la Cruz, y mis lectores el ing. Oswaldo Espinoza e ing. Javier Cóndor que me han ayudado en la realización de la disertación, dándome sus consejos para realizar y culminar una disertación de calidad.

Contenido

CAPITULO I: INTROD-UCCIÓN 8
1.1. Datos de la Organización 9
1.2. Alcance 9
1.3. Objetivos 10
1.3.1. Objetivo General 10
1.3.2. Objetivos Específicos 10
CAPÍTULO II: FUNDAMENTO TEÓRICO. 11
2.1 Metodología 11
2.1.1 Metodologías Estructuradas 12
2.1.2 Metodologías Orientadas a Objetos. 16
2.1.3 Metodologías Ágiles. 18
2.2 Selección de la metodología 20
2.3 Definición de arquitectura 21
2.4 Herramientas de desarrollo 23
CAPÍTULO III: CASO DE ESTUDIO. 25
1.Definición de requerimientos. 25
Descripción del sistema 25
Identificación de Usuarios Principales 26
Objetivos y alcance del sistema 26
Requerimientos funcionales. 26
Requerimientos no funcionales. 26
Diseño del sistema. 27
Diagrama Caso de uso Gestión de productos 28
Diagrama Caso de uso Gestión de Ordenes 36
Diagramas de Base de datos 43
Diagrama Conceptual 43
Diagrama Lógico 44
Diagrama Físico 45
Diagramas de Secuencia 45
Implementación y prueba. 52
Implementación 52
Pruebas 56

Funcionamiento y mantenimiento 60
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES 62
CAPÍTULO V: BIBLIOGRAFÍA 64

Tabla de ilustraciones

Ilustración 1:- Ciclo de Vida del Software 13
Ilustración 2 Fases de metodología RUP 17
Ilustración 3 Metodología SCRUM 19
Ilustración 4 Arquitectura tres capas 21
Ilustración 5 Arquitectura MVC 22
Ilustración 6: Código Index () y Details (string id) de Producto 52
Ilustración 7: Código HttppPost DeleteConfirmed (string id) 53
Ilustración 8: Pantalla Index Productos 53
Ilustración 9: Pantalla Editar Productos 54
Ilustración 10: Pantalla Detalles Producto 54
Ilustración 11: Pantalla Confirmación de eliminación producto 55
Ilustración 12: Pantalla index ordenes 55
Ilustración 13: Pantalla detalles de órdenes 56
Ilustración 14: Primera opción de instalación sistema 61
Ilustración 15 Segunda opción de instalación del sistema 61

Tabla de Diagramas

Diagrama 1: Diagrama General 27

Diagrama 2: Caso uso Gestión de Productos 28

Diagrama 3: Detalle caso de uso F1.1: Insertar producto 29

Diagrama 4: Detalle caso uso F1.2: Modificar Producto 30

Diagrama 5: Detalle caso uso F1.3: Eliminar Producto 32

Diagrama 6: Detalle caso uso F1.4.1: Búsqueda general Producto 33

Diagrama 7: Detalle caso uso F1.4.2: Búsqueda detalle Producto 34

Diagrama 8: Diagrama Caso de uso F2: Gestión de Ordenes 36

Diagrama 9: Detalle caso de uso F2.1: Insertar de Ordenes 36

Diagrama 10: Detalle caso de uso F2.2: Modificar Ordenes 38

Diagrama 11: Detalle caso de uso F2.3: Eliminar Ordenes 39

Diagrama 12: Detalle caso de uso F2.4.1: Búsqueda general Ordenes 40

Diagrama 13: Detalle caso de uso F2.4.2: Búsqueda Detalle Ordenes 41

Diagrama 14: Diagrama Conceptual de Base de datos 43

Diagrama 15: Diagrama lógico base de datos 44

Diagrama 16: Diagrama físico de base datos 45

Diagrama 17: Diagrama secuencia Ingreso Productos 46

Diagrama 18: Diagrama secuencia Modificar Productos 47

Diagrama 19: Diagrama secuencia Modificar Productos 47

Diagrama 20: Diagrama secuencia Búsqueda Productos 48

Diagrama 21: Diagrama secuencia Búsqueda Parámetros Productos 48

Diagrama 22: Diagrama secuencia Insertar Órdenes 49

Diagrama 23: Diagrama secuencia Modificar Órdenes 50

Diagrama 24: Diagrama secuencia Eliminar Órdenes 50

Diagrama 25: Diagrama secuencia Búsqueda Órdenes 51

Diagrama 26: Diagrama secuencia Búsqueda parámetros Órdenes 51

CAPITULO I: INTRODUCCIÓN

La presente disertación está basada en realizar una aplicación web usando técnicas procedimientos ingenieriles para el restaurante CHOCHOTE, que quiere implementar alta tecnología en la atención de sus clientes. Una observación que se hizo en los clientes es que algunos de ellos ya conocen los productos que el restaurante ofrece.

Con el desarrollo de esta aplicación se pretende gestionar las ordenes de sus clientes, atender de una manera más placentera a los clientes que ya conocen sus productos.

Para la implementación del sistema en la presente disertación se usará el ciclo de vida en cascada, esta metodología está orientada a objetos y permite realizar cambios cuando haga falta realizar correcciones en el sistema de manera muy sencilla, sin la utilización de iteraciones del proceso como es el caso de Scrum o proceso Racional Unificado.

Esta metodología está estructurada por varias fases, las cuales son: gestión de requisitos, análisis del sistema y diseño de aplicación, implementación del sistema, pruebas del sistema y despliegue. (Tinoco Gómez, Rosales López, & Bacalla, 2010)

El patrón de diseño utilizado es modelo vista controlador(MVC), que divide al proyecto en: el modelo, donde se realiza la conexión a la base de datos, relaciones entre tablas requeridas, restricciones de las variables; la vista se encuentra el código que va a conformar la interface para el usuario (interfaz gráfica) ; y finalmente el controlador, que son todas las funciones que permite comunicar entre la capa del modelo y la vista, aquí también se detallan funcionalidades requeridas para el control del sistema (Capa de reglas del negocio). (Microsoft, 2009)

Se ha dividido la presentación de la disertación en los siguientes capítulos: Capitulo I, dando una breve introducción e informando acerca de la situación actual de la empresa en la que se aplicara

el proyecto; Capitulo II, donde se presenta el marco teórico que nos da las bases requeridas para la disertación, describiendo las herramientas que se utilizarán, explicación del proceso de desarrollo entre otros; Capítulo III, presenta la aplicación del marco teórico, es decir el caso de estudio donde detallamos todo el proceso y entregamos el diseño con la documentación del sistema y el desarrollo del mismo; el Capítulo IV recoge las conclusiones y recomendaciones que surgieron durante el desarrollo del sistema ; finalmente el Capítulo V se encuentra la bibliografía.

1.1. Datos de la Organización

Nombre: CHOCHOTE.

Ubicación: Av. Jerónimo Carrión E 10-50 y Av. 12 de octubre Quito, Ecuador.

Actividad: Ofrecen una gran variedad de comidas rápidas como son: Cevichochos, pasteles, Jugos, Batidos, ensaladas de futas entre otros.

Características: Con 7 años de trayectoria el restaurante CHOCHOTE se caracteriza por su excelente atención y sobre todo sus típicas delicias culinarias, u principal fortaleza son sus originales recetas enfocadas en satisfacer a sus clientes

El horario de atención del restaurante es desde las 9:00 hasta las 17:00 de lunes a viernes.

1.2. Alcance

El presente trabajo finalizará con el diseño del sistema, desarrollo de la aplicación e implantación de un Sistema de Gestión de Ordenes en el restaurante CHOCHOTE.

El mismo que después de que los usuarios del sistema que lo administraran realicen las pruebas del sistema, lo aprueben para comprobar si cumple con los requisitos del cliente, solo en ese entonces se realizará la implantación de la aplicación culminara el proyecto.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema computacional en el restaurante CHOCHOTE mediante la utilización de arquitectura web que permita gestionar las órdenes de comida, brindando una mayor facilidad a sus clientes.

1.3.2. Objetivos Específicos

Analizar las características necesarias para cumplir con la gestión de realización de órdenes mediante la implementación del sistema.

Implementar los requerimientos para que el sistema funcione correctamente de acuerdo con las necesidades del restaurante CHOCHOTE y sus respectivos usuarios.

CAPÍTULO II: FUNDAMENTO TEÓRICO.

En este capítulo nos enfocamos en la investigación teórica que se debe realizar en varios tópicos relacionados con el modo de proceder en el trabajo de disertación.

Dentro del mismo explicaremos los diferentes tipos de metodologías que existen y se detallará que metodología se aplicará. En la presente disertación, se analiza las herramientas que se utilizarán para realizar el proyecto en los siguientes aspectos: modelado, el lenguaje de programación y base de datos.

2.1 Metodología

La metodología viene del griego metà (más allá), odòs (camino) y logos (estudio), en la informática es una recopilación de métodos, principios y reglas que permiten enfrentar un problema de manera sistemática y así desarrollar un sistema mucho más efectivo para las necesidades del cliente, las metodologías se estructuran con un inicio y un fin, en el cual el problema está resuelto. (Machala, 2010).

No se puede llamar a cualquier actividad metodología, debido a que no todas las actividades generalizan un gran número de ejemplos o actividades como son las metodologías, dentro de las aplicaciones en la Ingeniería del Software, se agrupan ciertos requisitos del cliente y se los generaliza para comprender que metodología nos conviene aplicar, logrando así un mejor entendimiento y resolución del problema. (Turner, 2015)

Las metodologías se clasifican en:

- Metodologías Estructuradas
- Metodologías Orientadas a Objetos
- Metodologías Agiles.

2.1.1 Metodologías Estructuradas

Metodologías también llamadas metodologías tradicionales, se enfocan en dividir el problema o sistema en partes más pequeñas de forma descendente "Top-Down" relacionadas entre sí, logrando enfatizar en las entradas y salidas de cada etapa del ciclo de vida del software. (Valladolid, 2009)

Entre su clasificación tenemos:

- Orientadas a procesos: Usando el método descendente de descomposición funcional para comprender los problemas del sistema transformándolos en requisitos dejando en su lugar un conjunto de diagramas que se conocen como especificación estructurada. (Valladolid, 2009)
- Orientadas a datos: Parte del modelo básico de obtener unas entradas, pasarlas al sector de procesamiento para al final obtener unas salidas, se enfocan en las entradas y salidas. En las cuales se procesan primeramente las estructuras de datos a continuación de las estructuras de control que deben ser jerárquicas junto a los componentes procedimentales que se deberán ajustar a dicha estructura. (Valladolid, 2009)

Entre sus características tenemos:

- Gran volumen de datos.
- Abarca varias ideas de la organización.
- Larga duración de desarrollo.

Una de las principales metodologías estructuradas es la que se basa en modelo en cascada, que se presenta a continuación.

2.1.1.1 Modelo en cascada

Este modelo es uno de los primeros publicado a partir de procesos de ingeniería mucho más generales, se lo conoce como cascada por la transición de una etapa a otra, se las realiza una a la vez y en un orden establecido como se presenta en la ilustración 1. (Sommerville, 2011)

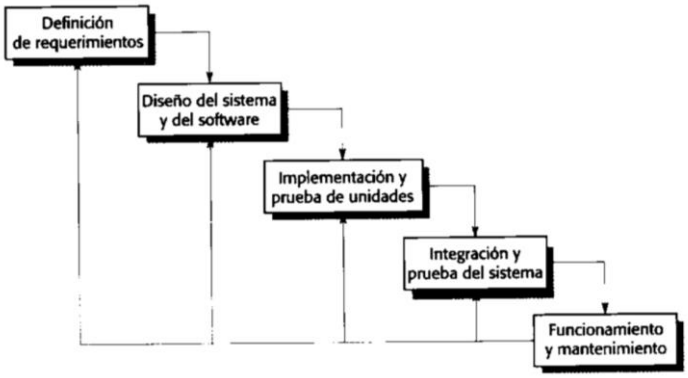


Ilustración 1: Ciclo de Vida del Software

En la Ilustración 1 se presenta el ciclo de vida del modelo de cascada, el cual fue la guía para la construcción de Software por más de 20 años.

La ventaja del modelo en cascada es que se produce documentación en cada fase y se logra coincidir con otros procesos de ingeniería. (Sommerville, 2011)

Una desventaja del proceso es que se vuelve imposible la división del proceso en varias etapas, además se declaran los requerimientos en el inicio del ciclo de vida, lo que complica el ambiente si se requieren realizar cambios durante el desarrollo. (Sommerville, 2011)

El modelo en cascada se debe implementar en sistemas que se conozca bien los requerimientos, sea improbable que exista un cambio o cambien los requerimientos durante el desarrollo del sistema. (Sommerville, 2011)

De acuerdo con la ilustración 1, las principales fases son:

- Análisis y definición de requerimientos: Se consulta con el usuario los servicios, metas y restricciones, se formulan y especifican a detalle y se convierten en requerimientos y especificaciones del sistema. (Sommerville, 2011)
- Diseño del sistema y del Software: Divide los requerimientos del sistema en dos importantes componentes, hardware y software. Se detalla a nivel técnico que componentes se realizarán y cuáles son los que deben entrar en cada división. (Sommerville, 2011)
- Implementación y prueba de unidades: Durante esta etapa se desarrolla como un solo conjunto o por unidades del mismo, produciendo así un conjunto de pruebas de unidad que sirve para verificar si se lograron cumplir los requerimientos según se especificó. (Sommerville, 2011)
- Integración y pruebas de sistema: Los programas o unidades individuales del sistema se prueban e integran como un solo sistema completo, asegurando así que se cumple los requisitos solicitados, una vez probado el producto se entrega al cliente. (Sommerville, 2011)
- Funcionamiento y mantenimiento: Por lo general esta es la etapa más larga del ciclo de vida, consiste en poner en funcionamiento el sistema. El mantenimiento es en cambio para corregir posibles errores que no se detectaron en etapas anteriores del ciclo de vida, mejorar los servicios del sistema y descubrir nuevos requerimientos. (Sommerville, 2011)

Entregables

Cada etapa del ciclo de vida tiene como resultado documentos entregables que deben ser firmados por el cliente, para avanzar a la siguiente etapa se debe primero finalizar la anterior, entre etapas pueden existir varias iteraciones que son costosas para el producto final, no es un proceso lineal, pero las iteraciones deben ser pocas, un ejemplo de iteraciones es que se encuentren errores en el diseño con los requerimientos, por lo tanto regresamos y corregimos para lograr avanzar a la siguiente etapa. (Sommerville, 2011)

Entregables en cada fase:

1. **Definición y requerimientos**. En esta fase se entrega una lista de los requerimientos del usuario, para saber cuáles son las necesidades puntuales del usuario a desarrollar. En este proceso se detallan: Actores, Casos de Uso y procesos detallados (SRD: documento de especificación de requisitos).
2. **Diseño del sistema y del software.** Se entregan dos documentos, el primero conteniendo la arquitectura del sistema y el segundo que contiene el diseño detallado. (SDD: Documento de diseño del software).
3. **Implementación y prueba de unidades.** Se entrega la codificación del sistema, con sus respectivos módulos.
4. **Integración y pruebas del sistema.** Se entregan las pruebas de aceptación, de módulos, de integración y de sistema.
5. **Funcionamiento y mantenimiento.** Se encarga de encontrar errores en el sistema que no se encontraron en etapas anteriores, además puede mejorar el uso del sistema encontrando nuevos requerimientos adema de que puede llegar a ser la fase más larga del ciclo de vida.

2.1.2 Metodologías Orientadas a Objetos.

Las metodologías orientadas a objetos se enfocan en el desarrollo de sistemas que son dirigidos a reutilizar su código, logrando así usar objetos ya existentes en la aplicación y las facilidades de compartir componentes entre sistemas, un objetivo principal es construir aplicaciones nuevas uniendo componentes que ya existen. (Officer, 2009)

Beneficios de adoptar la metodología Orientada a Objetos

- Mejorar la productividad al reusar componentes existentes.
- Manejar la complejidad rompiendo paradigmas complejos en pequeños componentes encapsulados.

Consideraciones para adoptar una metodología Orientada a Objetos:

- Proyectos de mediano y gran tamaño: Cuando el sistema es de gran tamaño se descompone en varios componentes, por lo tanto, en grandes proyectos es una gran solución.

Dentro de las metodologías orientadas a objetos, una de las más representativas es la metodología RUP (proceso Racional Unificado) que se presenta a continuación.

2.1.2.1 Metodología RUP (Proceso Racional Unificado)

Divide el proceso de desarrollo en varias fases iterativas, dentro de sus características se encuentran:

- Manejo de requerimientos: Describe como la organización debe seguir los pasos de los requerimientos funcionales. En la metodología además especifica que son una guía hacia

el diseño, implementación y prueba, siendo un elemento totalmente integrador. (Houston, 2002)

- Arquitectura basada en uso de componentes: Hace enfoque en realizar los componentes que pueden ser reutilizados, permitiendo una visión clara del sistema resaltando sus componentes más importantes, lo cual nos facilita mucho más el desarrollo.
- Control de cambios del software: Describe como se deben manejar los cambios a lo largo del desarrollo. (Powell-Morse, 2017)
- La arquitectura: Va muy a la par junto con los casos de uso como se indica en la ilustración 2, esto requiere que la metodología permita la constante evolución de los dos componentes durante todo el proceso. (Houston, 2002)

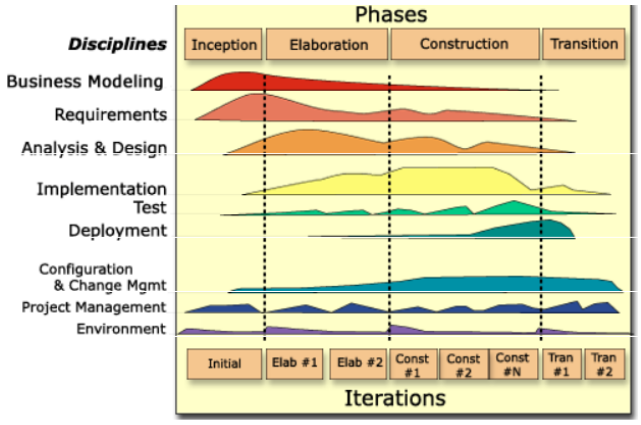


Ilustración 2: Fases de metodología RUP

- Dentro de la Ilustración 2 podemos encontrar sus fases descritas por inicio, elaboración, construcción y transición. (Houston, 2002)
- El objetivo de la fase de inicio es estar en contacto con todos los involucrados del proyecto, es mucho más larga esta fase si se va a empezar un nuevo proyecto, lo contrario a cuando se está realizando una continuación de proyectos antiguos. (Houston, 2002).

- El objetivo de la fase de elaboración es principalmente dar forma a la arquitectura que se va a implementar y junto con el diseño pasar a la construcción del proyecto. (Houston, 2002)
- El objetivo de la fase de construcción es entender todos los requerimientos restantes y completar el ciclo de desarrollo basad en una línea arquitectónica. (Houston, 2002)
- El objetivo de la fase de transición es asegurarse que su software esté disponible para el usuario final, la fase puede tener varias iteraciones. (Houston, 2002)

2.1.3 Metodologías Ágiles.

Las metodologías agiles se enfocan en la satisfacción del cliente, el cual se involucra y compromete con el desarrollo, dedicándose a tenerlo informado del avance del proyecto constantemente, además logrando así aportar con experiencia de parte de los desarrolladores y del cliente evitando malentendidos al final del desarrollo. (Radigan, 2017)

Una de las metodologías Ágiles más conocidas y usadas es Scrum, la cual se la presenta a continuación. Dentro de sus principales características se destaca por ser orientada a un grupo de desarrollo dividiendo las tareas.

2.1.3.1 Scrum

Scrum es una metodología Ágil que crea una serie de iteraciones de longitud fija llamadas Sprint que da un progreso fijo en la finalización de cada iteración como lo muestra a continuación la ilustración 3, en cada sprint se divide en diferentes fases:

- Planificación de sprint: reunión de planificación del equipo para determinar que hay que hacer en el siguiente sprint.

- Reuniones diarias: se pone al tanto al equipo de software de las dificultades y progresos, expresados rápidamente para continuar con el trabajo o resolver conflictos en las tareas asignadas.
- Demostración del sprint: se enseña lo que completaron en la iteración, se trata de una reunión informal con el cliente.
- Retrospectiva del sprint: se revisa lo que fue bien, mal y lo que hay que mejorar en el equipo.

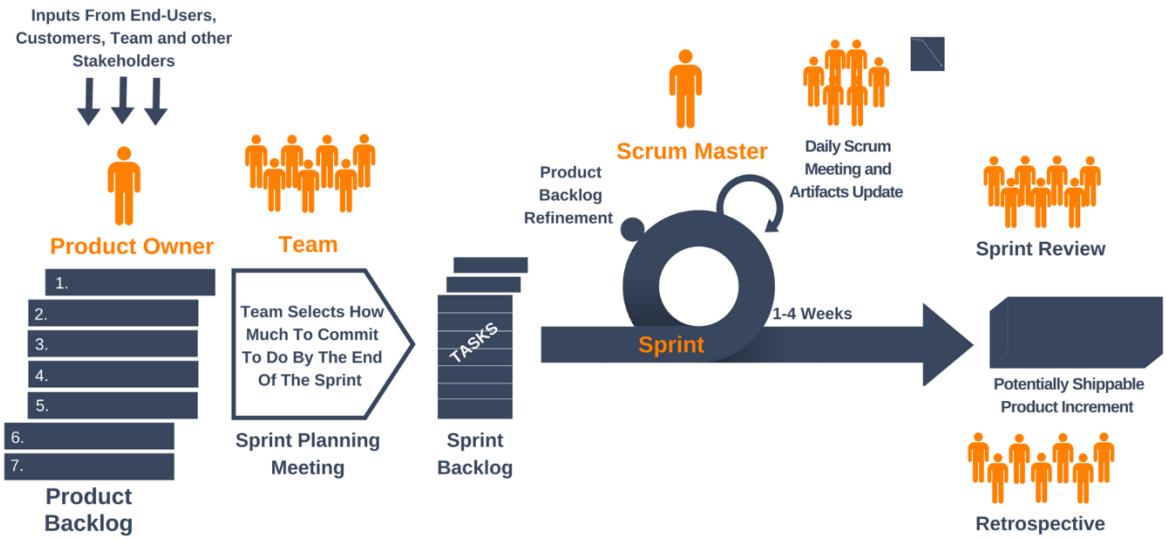


Ilustración 3: Metodología SCRUM

- La ilustración 3 nos indica cómo se maneja aplicación de la metodología SCRUM, el cual empieza con las entradas que son los requerimientos del cliente final (inputs from end users), los cuales el product owner está en cargado de resolver, el product owner las clasifica en requerimientos por cumplir que se convierte en el product backlog, se reúne

junto a su equipo para que se seleccionen tareas del product backlog que se trataran de resolver en el sprint, lo cual llega a conformar el sprint backlog, el equipo está dirigido por un Scrum Master que planifica los sprints, refina requerimientos y brinda apoyo a su equipo. El Scrum Master se mantiene en constante comunicación con el product owner además de mediar y representar al equipo, cada sprint es planeado para que tenga una duración de 1 a 4 semanas y después de que se cumple el tiempo del sprint (lo cual es un periodo relativamente corto de tiempo), se analiza que se hizo bien y mal en el sprint que finalizo (Sprint Review), para que sirva como referencia para el siguiente sprint planteando que se mejorara en el futuro (Retrospective), se vuelve a planificar otra iteración y así sucesivamente hasta completar todos los requisitos del Product owner. (Francia, 2017)

2.2 Selección de la metodología

Para el presente proyecto se elegirá la metodología en cascada debido a los siguientes motivos que se especifican a continuación.

- Se tiene el proceso bien definido en el caso de estudio, debido a que se ha realizado el proceso muchas veces y se conocen todos los pasos a seguir.
- Se requiere solamente un solo ciclo de desarrollo
- El dominio del problema es conocido.
- La interacción con el cliente no es tan necesaria, debido a que se conoce el dominio del problema y se tiene bien definido el proceso.

2.3 Definición de arquitectura

La primera parte en la que debemos enfocarnos en este concepto es para que no sirve aplicar o establecer una arquitectura de aplicación, lo cual nos lleva directamente a una respuesta clara, la arquitectura nos orienta para entender cómo debe organizarse el sistema. (Sommerville, 2011)

Siendo la principal conexión entre el diseño y la ingeniería de requerimientos ya que identifica los componentes del sistema y la relación que existe entre ellos.

2.3.1 Arquitectura en tres capas

A continuación, la ilustración 4 nos señala cómo se relacionan las capas, a más capas es considerado más robusto el sistema, ya que en cada capa se controla el flujo de datos:

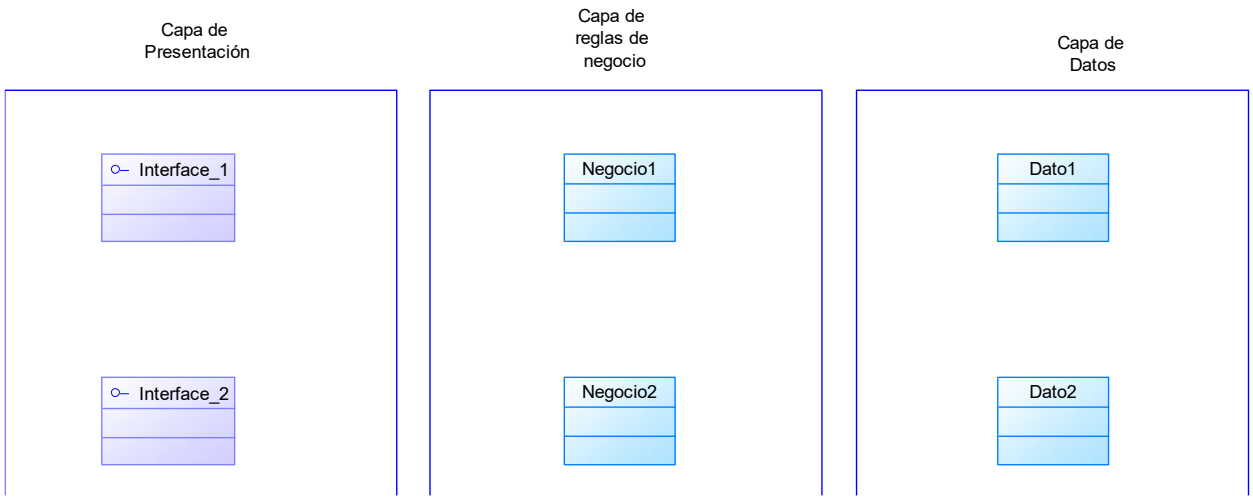


Ilustración 4: Arquitectura tres capas

La arquitectura en tres capas es muy utilizada en aplicaciones que siguen un modelo de negocio, toda aplicación tiene un objetivo acceder a una base de datos, usar los datos y

presentarlos de manera deseada, además se debe procesar e imponer reglas de uso de los datos siendo estas llamadas las reglas del negocio. Este tipo de arquitectura hace que se comuniquen estas tres áreas entre sí. (Gómez, n.d.)

2.3.1 Patrón de arquitectura Modelo Vista Controlador(MVC).

La ilustración 5 nos indica que es un modelo arquitectónico el cual separa la aplicación en tres componentes principales: el modelo, la vista y el controlador. Usado en su mayoría para aplicaciones web.

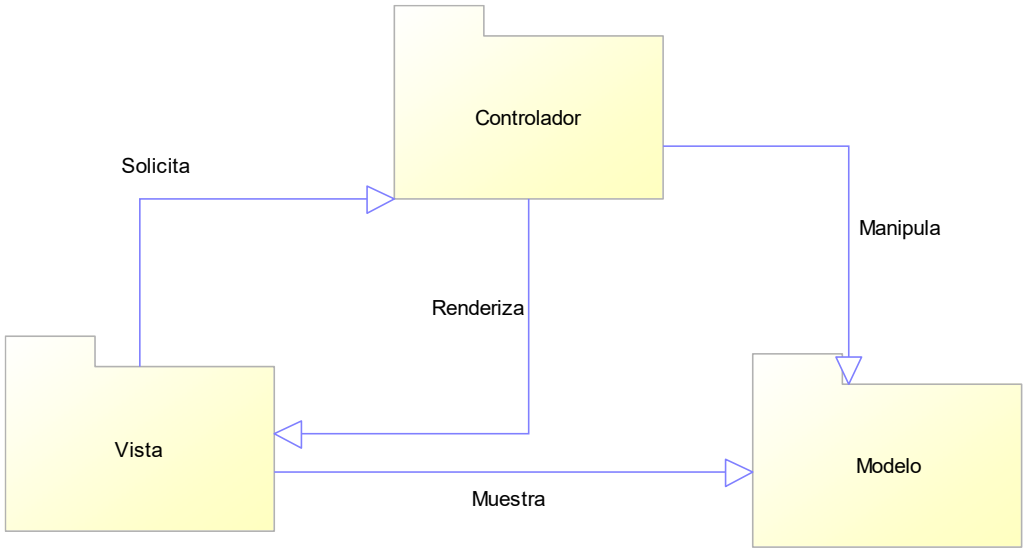


Ilustración 5: Arquitectura MVC

Modelo: En esta sección se controla la lógica del dominio de datos, manipulan y recuperan datos desde la base de datos, recuperan información y trabajan con ella, se pueden tener varios modelos con varias bases de datos.

Vista: Son los componentes que muestran la interfaz de usuario, en su mayoría todos los datos para crear la vista se obtienen de los modelos.

Controlador: Controlan la interacción del usuario, deciden que vista se debe presentar según las instrucciones solicitadas y trabajan con los modelos.

2.4 Herramientas de desarrollo

Modelado

Consiste en realizar modelos abstractos de un sistema, de donde se van conformando diferentes vistas del problema a resolver, en este caso es el sistema que se debe desarrollar. (Sommerville, 2011)

Power Designer

Herramienta que nos permite realizar el modelado gráfico del sistema, mediante diagramas orientados a objetos representaremos diferentes aspectos del mismo, usaremos diagramas UML (Lenguaje de modelado unificado) para el entendimiento interno del sistema y pasar a la siguiente etapa que es el desarrollo en si del sistema. (SAP, 2018)

Desarrollo

En esta Etapa se implementa el software ejecutable del sistema, el cual seguirá las guías y directrices de la etapa anterior que fue el diseño.

Microsoft Visual Studio IDE (Ambiente de desarrollo integrado)

El ambiente de desarrollo permite desarrollar en lenguaje C# que es un lenguaje orientado a objetos, permite desarrollo orientado a la web como el patrón de diseño que se eligió (MVC) lo cual nos entrega una total utilidad para este caso de estudio, además el cliente se encuentra actualmente con sistema operativo Windows lo cual nos hace seguir la misma línea de productos a pesar de que el producto final será para cualquier plataforma. (Microsoft, 2009)

SQL Server

Es un Sistema de manejo de bases de datos relacionales desarrollado por la empresa Microsoft uno de los más populares en la actualidad, nos permite ejecutar varias sesiones en un mismo servidor físico lo cual la hace un gestor de base de datos muy robusto que permite realizar el trabajo como cliente servidor, en donde el cliente no almacena datos, sino los envía y consulta del servidor.

CAPÍTULO III: CASO DE ESTUDIO.

En el caso de estudio de la presente disertación aplicaremos la metodología en cascada, debido a que sus requerimientos son muy poco probables que tiendan a cambiar, se realizará una sola iteración y se describirán las fases y procedimientos para cumplir con la metodología establecida.

Descripción de la situación actual de la empresa

CHOCHOTE se está manejando actualmente por un sistema de facturación, pero dentro de sus nuevos requerimientos se encuentra el desarrollar un gestor de órdenes para facilitar su atención a los clientes, ya que actualmente se encuentra realizando la gestión de las órdenes en papeles individuales.

La meta es lograr administrar de una forma más adecuada las órdenes que tienen, y por sobre todo dar una manera más cómoda a sus clientes para que elijan sus productos.

1.Definición de requerimientos.

Descripción del sistema

El sistema es una aplicación que permitirá el manejo de las órdenes de los clientes, ya sea que ellos ordenen sus productos de manera virtual o se los ingrese con los administradores del sistema interactuando con el cliente, facilitando así la realización de las órdenes y su gestión para que al momento en que se pasen al siguiente paso del proceso que es la elaboración de los platos que se requieren.

Dentro de las expectativas del cliente están extender su sistema a una aplicación móvil que genere las ordenes listas para su pago.

Identificación de Usuarios Principales

Administrador del sistema: Es el usuario que tiene acceso permitido para gestionar los productos, esto incluye agregar, modificar, eliminar, y buscar productos por sus parámetros.

Cliente: Es el usuario final del sistema, el que requerirá los productos que se desea sean enviados a preparar.

Objetivos y alcance del sistema

Se requiere implementar una nueva forma de que nuestros clientes realicen sus órdenes, lo cual requiere que el sistema cumpla los requerimientos del establecimiento para brindar una nueva alternativa para los clientes más frecuentes.

Requerimientos funcionales.

F0: El sistema permitirá crear su respectivo usuario a cada cliente

F1: El sistema permitirá gestión de productos.

F2: El sistema permitirá gestión de órdenes.

F3: El sistema permitirá generar reportes.

F4: Gestión de usuarios.

Requerimientos no funcionales.

NF1: El sistema será multiplataforma

NF2: El sistema será fácil de operar

Diseño del sistema.

Diagrama General: Es un diagrama representando al sistema, el cuadrilátero representa a lo contenido en el sistema, las circunferencias representan a las funcionalidades del sistema como se lo indica a continuación en el Diagrama 1. (Sommerville, 2011)

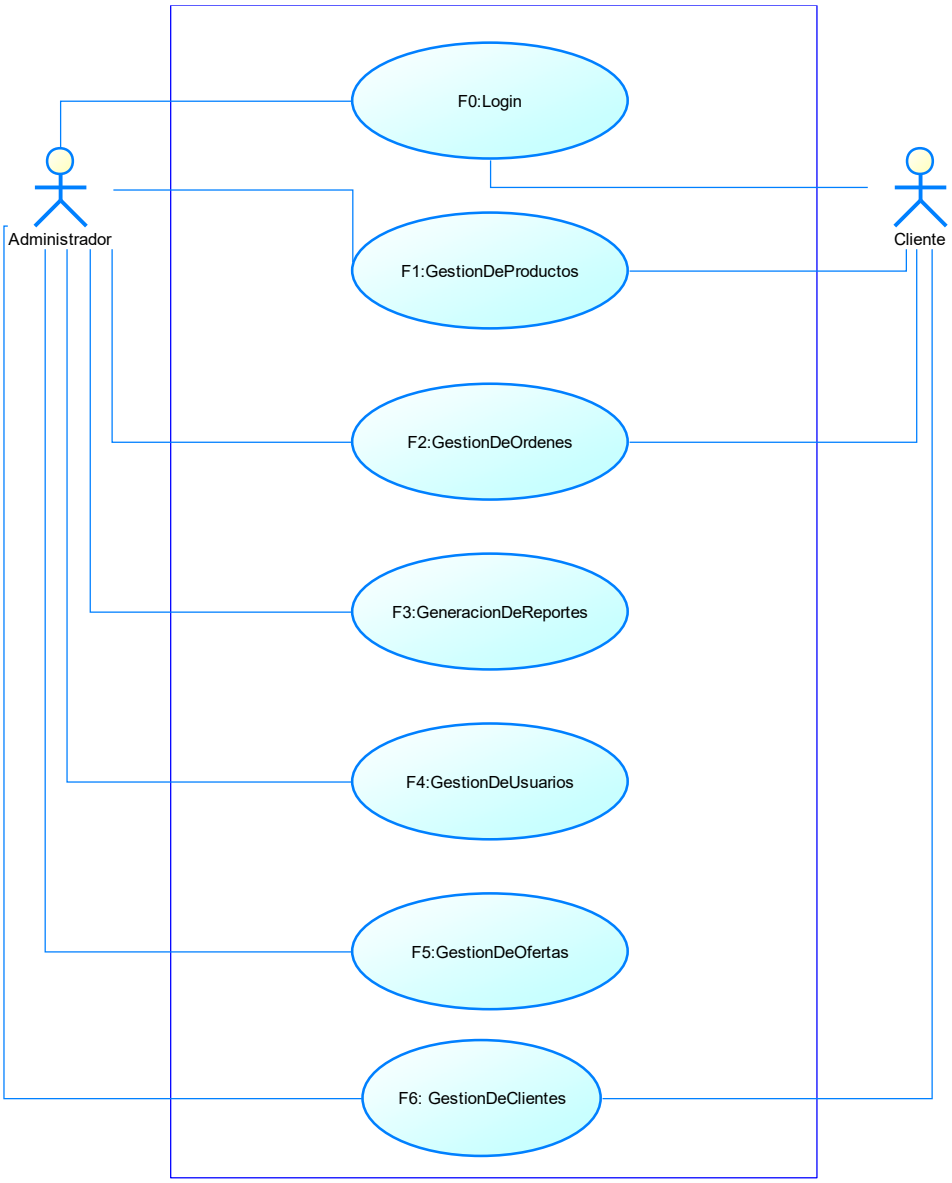


Diagrama 1: Diagrama General

Diagrama de casos de uso: Son diagramas de modelado que se usan en la representación del sistema, y también sirven para apoyar la adquisición de requerimientos como se presenta a continuación en el Diagrama 2. (Sommerville, 2011)

Por cada caso de uso se profundizará en su funcionalidad, tomando en cuenta que solo se presentara Ordenes y Productos, el resto de diagramas se encuentran adjuntos en el Anexo en el CD que se entrega.

Diagrama Caso de uso Gestión de productos

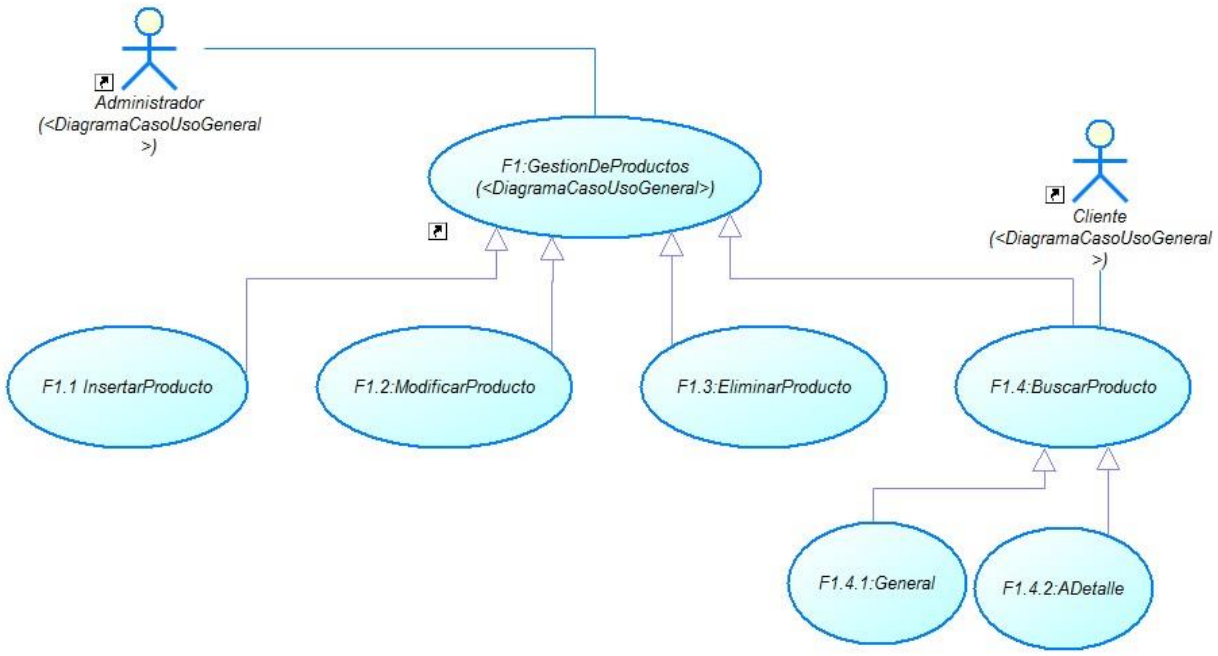


Diagrama 2: Caso uso Gestión de Productos

Detalle de caso de uso: Permite describir los componentes y requisitos en mínimo detalle de una funcionalidad, para así establecer la claridad de los requerimientos, estos diagramas deben ser aprobados por el cliente para comenzar a desarrollar. (Sommerville, 2011)

Por cada caso de uso se avanzará un nivel más hacia el diagrama a detalle, lo cual nos da una idea más detallada de los componentes de la funcionalidad.

Detalle caso de uso Gestionar Producto:

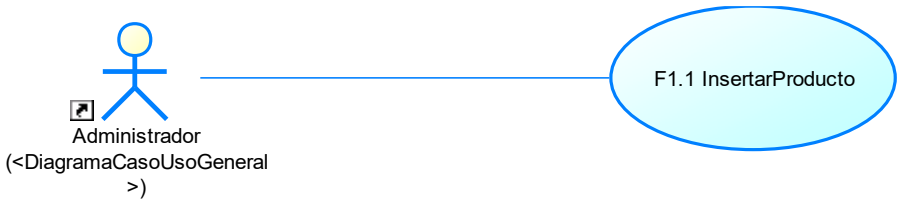


Diagrama 3: Detalle caso de uso F1.1: Insertar producto

F1.1	Insertar Producto	
Descripción	El sistema permite al actor ingresar nuevos productos, los cuales contendrán los campos necesarios para identificarlo en la base de datos.	
Actores:	Administrador	
Flujo principal	**Paso**	**Acción**
	1	El actor selecciona la opción Ingresar Producto.
	2	El sistema ingresa a la pantalla para ingresar un producto.
	3	El actor ingresa los datos del producto.
	4	El sistema comprueba y valida los datos ingresados. (E1)

	5	El sistema Almacena en la base de Datos. (E2)
Flujo alterno	**Paso**	**Acción**
	4	Si el producto ya existe ver caso de uso F1.1: Modificar Producto, F1.3: Eliminar Producto
Excepciones	**Código**	**Acción**
	E1	El producto ya existe
	E2	El sistema no logró conexión a la base de datos

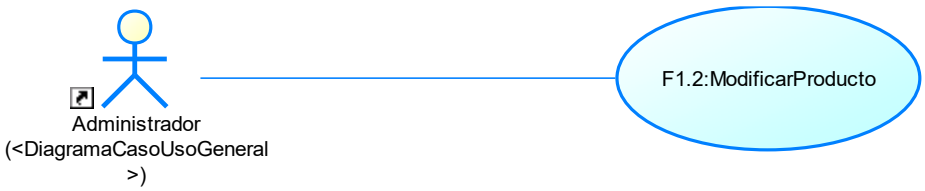


Diagrama 4: Detalle caso uso F1.2: Modificar Producto

F1.2	**Modificar Producto**	
Descripción	El sistema permite modificar o corregir detalles de un producto que ya ha sido ingresado.	
Actores:	Administrador	
Precondición	El sistema debe tener ingresado al menos un producto.	
Flujo Principal	**Paso**	**Acción**

	1	El actor ingresa a la opción modificar producto.
	2	El sistema carga los productos existentes. (E1)
	3	El actor selecciona un producto.
	4	El sistema carga el producto
	5	El actor modifica o corrige la información del producto seleccionado.
	6	El actor guarda la información del producto. (E2)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

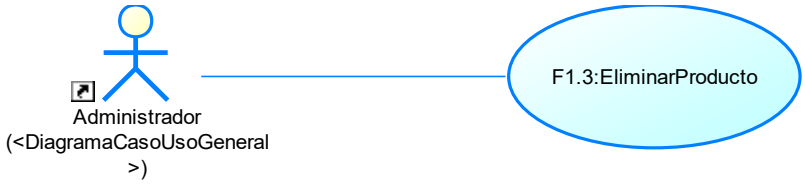


Diagrama 5: Detalle caso uso F1.3: Eliminar Producto

F1.3	**Eliminar Producto**	
Descripción	El sistema permite eliminar productos de la base de datos.	
Actores:	Administrador	
Precondición	El sistema debe tener ingresado al menos un producto.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción eliminar producto.
	2	El sistema despliega los productos existentes. (E1)
	3	El actor selecciona eliminar el producto
	4	El sistema despliega confirmación. (E2)
	5	El sistema elimina el producto.
Flujo alterno	**Paso**	**Acción**

Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

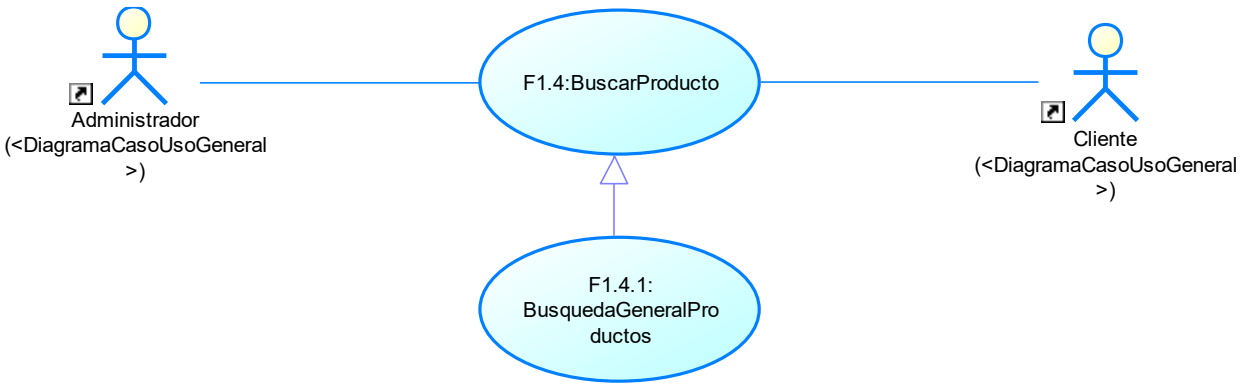


Diagrama 6: Detalle caso uso F1.4.1: Búsqueda general Producto

F1.4.1	**Búsqueda general de productos**	
Descripción	El sistema permite buscar productos de la base de datos.	
Actores:	Administrador, Cliente.	
Precondición	El sistema debe tener ingresado al menos un producto.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción de búsqueda

	2	El sistema carga los productos existentes. (E1)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.

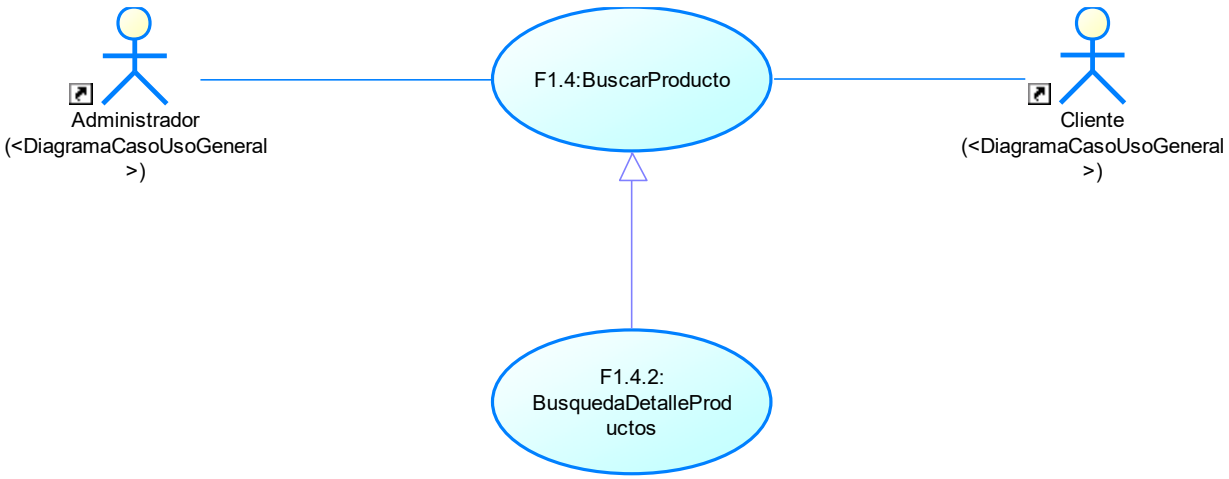


Diagrama 7: Detalle caso uso F1.4.2: Búsqueda detalle Producto

F1.4.2	**Búsqueda a detalle de producto**
Descripción	El sistema permite buscar un producto por medio de sus parámetros.
Actores:	Administrador, Cliente

Precondición	El sistema debe tener ingresado al menos un producto.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción de búsqueda
	2	El sistema carga los productos existentes. (E1)
	3	El actor ingresa parámetros de un producto.
	4	El sistema carga los productos existentes. (E2)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

Diagrama Caso de uso Gestión de Ordenes

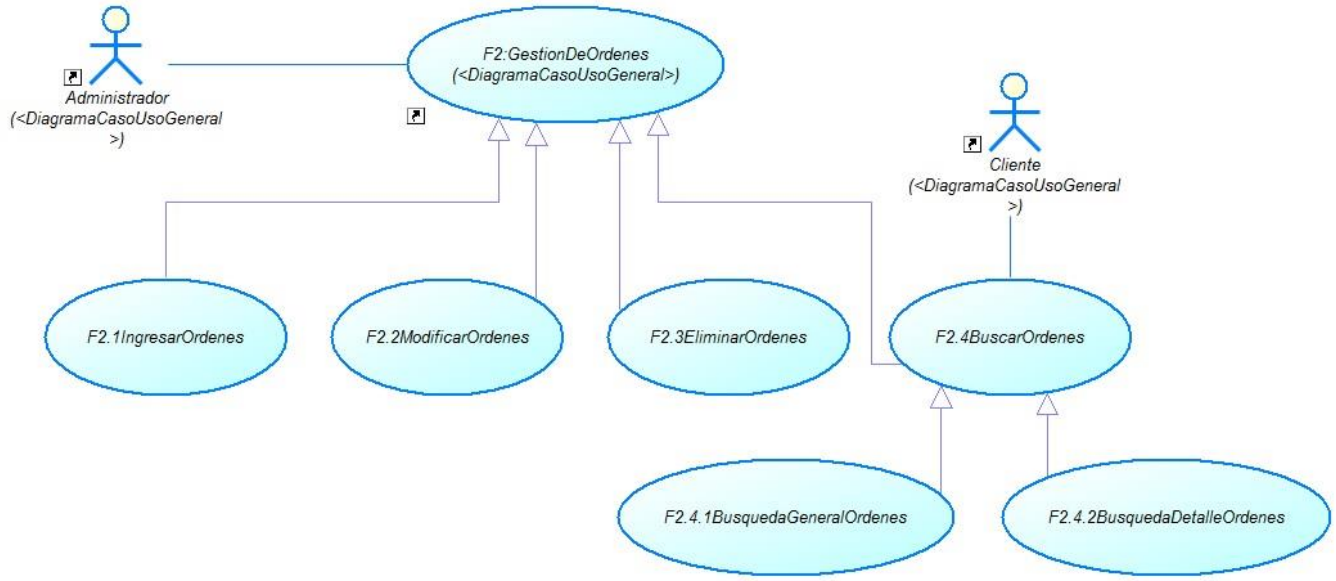


Diagrama 8: Diagrama Caso de uso F2: Gestión de Ordenes

Detalle caso de uso Gestión de Ordenes



Diagrama 9: Detalle caso de uso F2.1: Insertar de Ordenes

F2.1	**Insertar Ordenes**	
Descripción	El sistema permite al actor ingresar nuevas órdenes, las cuales contendrán los productos que son requeridos para preparar.	
Actores:	Administrador	
Flujo principal	**Paso**	**Acción**

	1	El actor selecciona la opción Ingresar Ordenes.
	2	El sistema presenta la pantalla desplegando los productos. (E1)
	3	El actor selecciona los productos.
	4	El actor selecciona ingresar la orden.
	5	El sistema Almacena la orden.
	6	El sistema despliega confirmación. (E2)
Flujo alterno	**Paso**	**Acción**
	4	Si el producto deseado no existe revisar caso de uso F1.1 Ingresar producto o comprobar inventario.
Excepciones	**Código**	**Acción**
	E1	Error de conexión con la base de datos

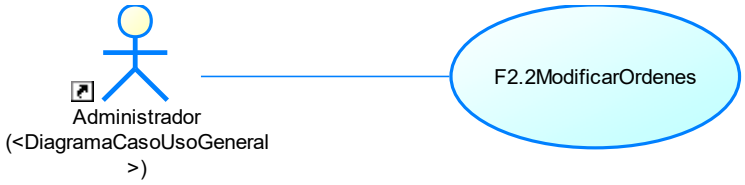


Diagrama 10: Detalle caso de uso F2.2: Modificar Ordenes

F2.2	**Modificar Ordenes**	
Descripción	El sistema permite modificar o corregir detalles de una orden ingresada a la base de datos.	
Actores:	Administrador	
Precondición	El sistema debe tener ingresado al menos un producto.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción modificar Orden.
	2	El sistema despliega las ordenes pendientes. (E1)
	3	El actor selecciona una orden.
	4	El actor modifica o corrige la información de la orden seleccionada.
	5	El actor guarda la información de la orden. (E2)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**

	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

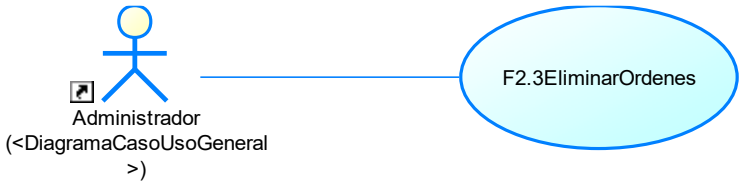


Diagrama 11: Detalle caso de uso F2.3: Eliminar Ordenes

F2.3	**Eliminar Ordenes**	
Descripción	El sistema permite eliminar órdenes de la base de datos.	
Actores:	Administrador	
Precondición	El sistema debe tener ingresado al menos una orden.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción eliminar orden.
	2	El sistema despliega las órdenes existentes. (E1)
	3	El actor selecciona una orden.
	4	El actor selecciona eliminar la orden.
	5	El sistema despliega confirmación. (E2)

	6	El sistema Elimina la orden.
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

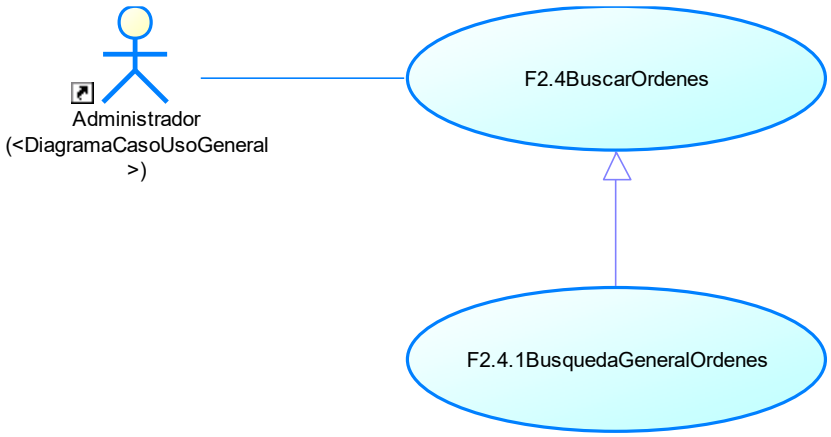


Diagrama 12: Detalle caso de uso F2.4.1: Búsqueda general Ordenes

F2.4.1	**Búsqueda general de Ordenes**
Descripción	El sistema permite buscar órdenes de la base de datos.

Actores:	Administrador	
Precondición	El sistema debe tener ingresado al menos un orden.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción de búsqueda de ordenes
	2	El sistema despliega las órdenes existentes. (E1)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.

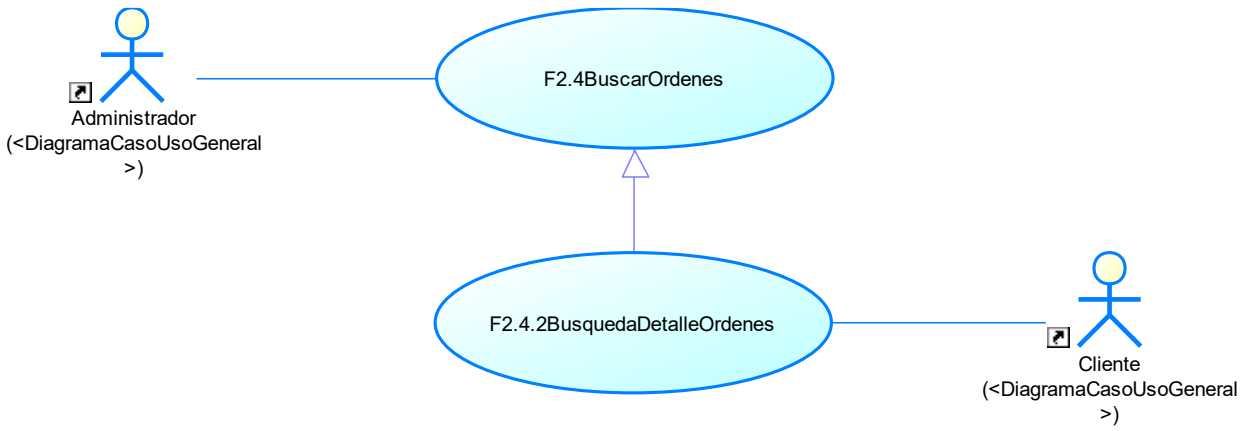


Diagrama 13: Detalle caso de uso F2.4.2: Búsqueda Detalle Ordenes

F2.4.2	**Búsqueda a detalle de una Orden**	
Descripción	El sistema perite buscar una orden por medio de sus parámetros.	
Actores:	Administrador, Cliente	
Precondición	El sistema debe tener ingresado al menos una orden.	
Flujo Principal	**Paso**	**Acción**
	1	El actor ingresa a la opción búsqueda de órdenes.
	2	El sistema despliega las órdenes existentes. (E1)
	3	El actor ingresa un parámetro de una orden.
	4	El sistema despliega las órdenes buscadas. (E2)
Flujo alterno	**Paso**	**Acción**
Excepciones	**Código**	**Acción**
	E1	Fallo de la conexión.
	E2	El sistema no logró conexión a la base de datos.

Diagramas de Base de datos

El diseño de un base de datos trata de descomponer el problema en pequeñas partes, para lograr así entender el problema del sistema que se trata de resolver. Así mismo se descompone los diagramas en conceptual, lógico y físico. (Alegria, 2012)

Diagrama Conceptual

El diagrama conceptual nos ayuda a entender lo más básico de la base de datos, es un paso fundamental para entender el problema, ya que el nivel de abstracción es muy alto. En este diagrama se permiten relaciones muchos a muchos para denotar que se necesita más abstracción en el diseño como nos muestra el diagrama 23 a continuación el cual es el diagrama conceptual final del sistema: (Sommerville, 2011)

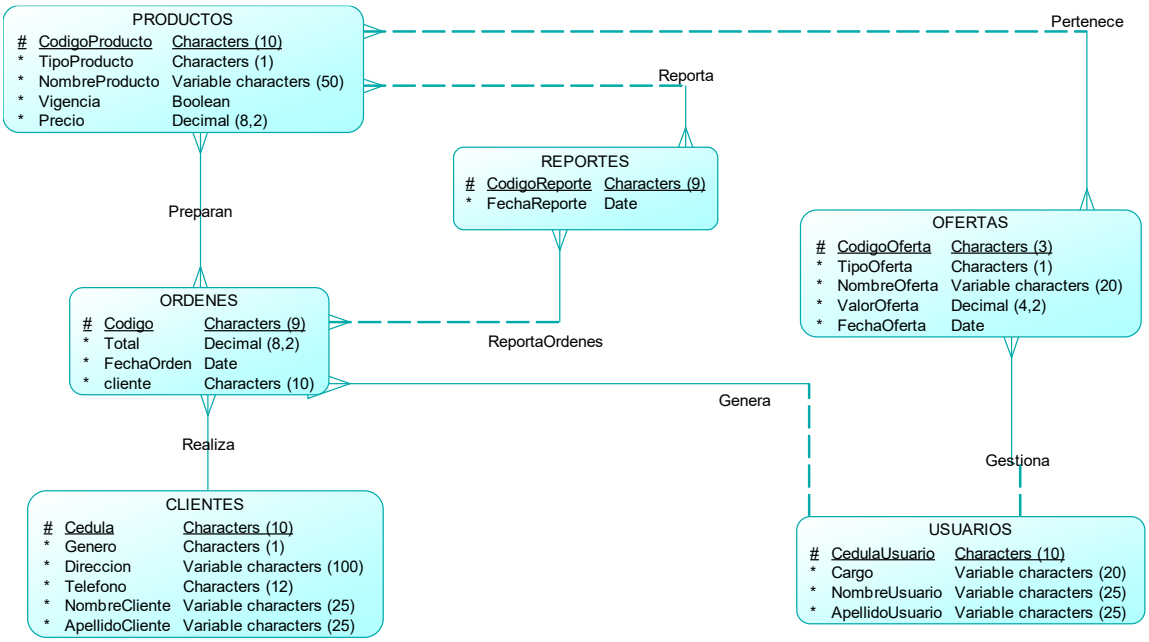


Diagrama 14: Diagrama Conceptual de Base de datos

Diagrama Lógico

Se realiza una menor abstracción de la base de datos, desglosando los enlaces muchos a muchos que se tenía, también se lo realiza para que lo entienda un sistema gestor de base de datos, que se lo elije de acuerdo con la necesidad del sistema como nos muestra el diagrama 24 a continuación: (Alegria, 2012)

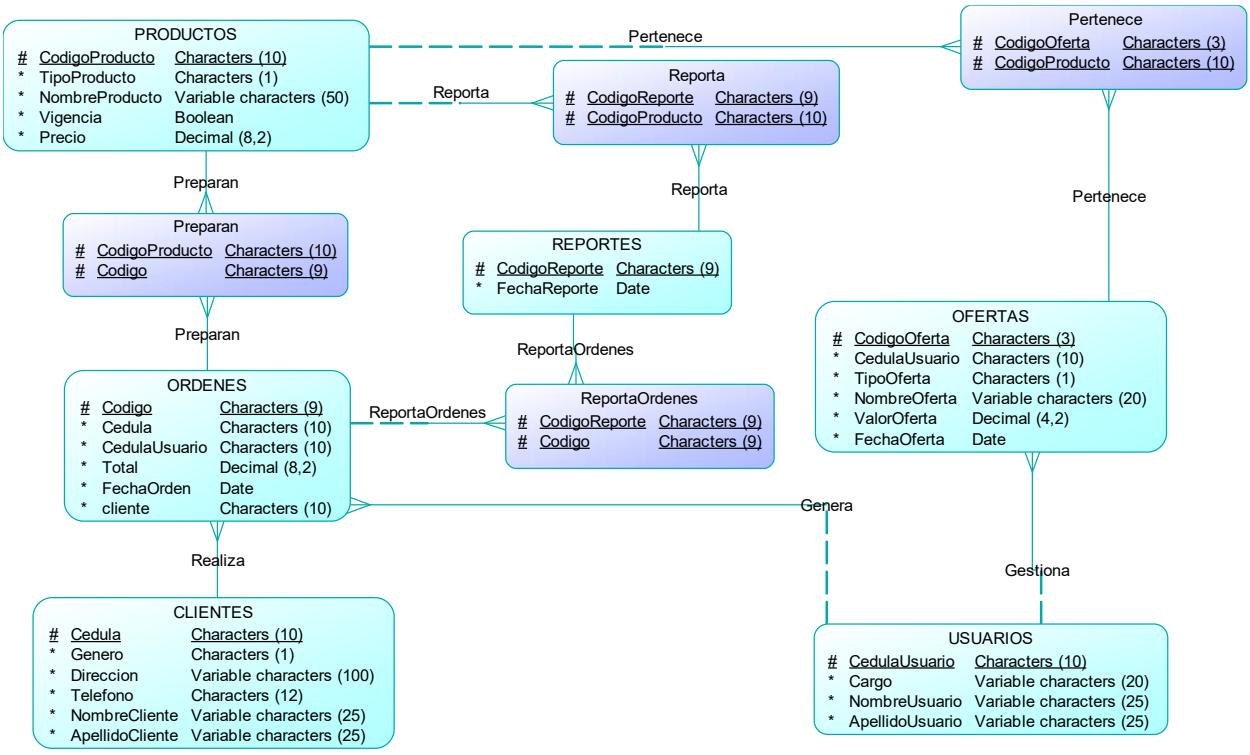


Diagrama 15: Diagrama lógico base de datos

Diagrama Físico

Se refina el diagrama lógico y se lo implementa en el sistema gestor de base de datos(SGBD), describe la implantación en una memoria secundaria y se realiza uno específico para el SGBD que lo utilizará. (Alegria, 2012)

A continuación, se presenta el diagrama físico de la base de datos del sistema en la ilustración 25:

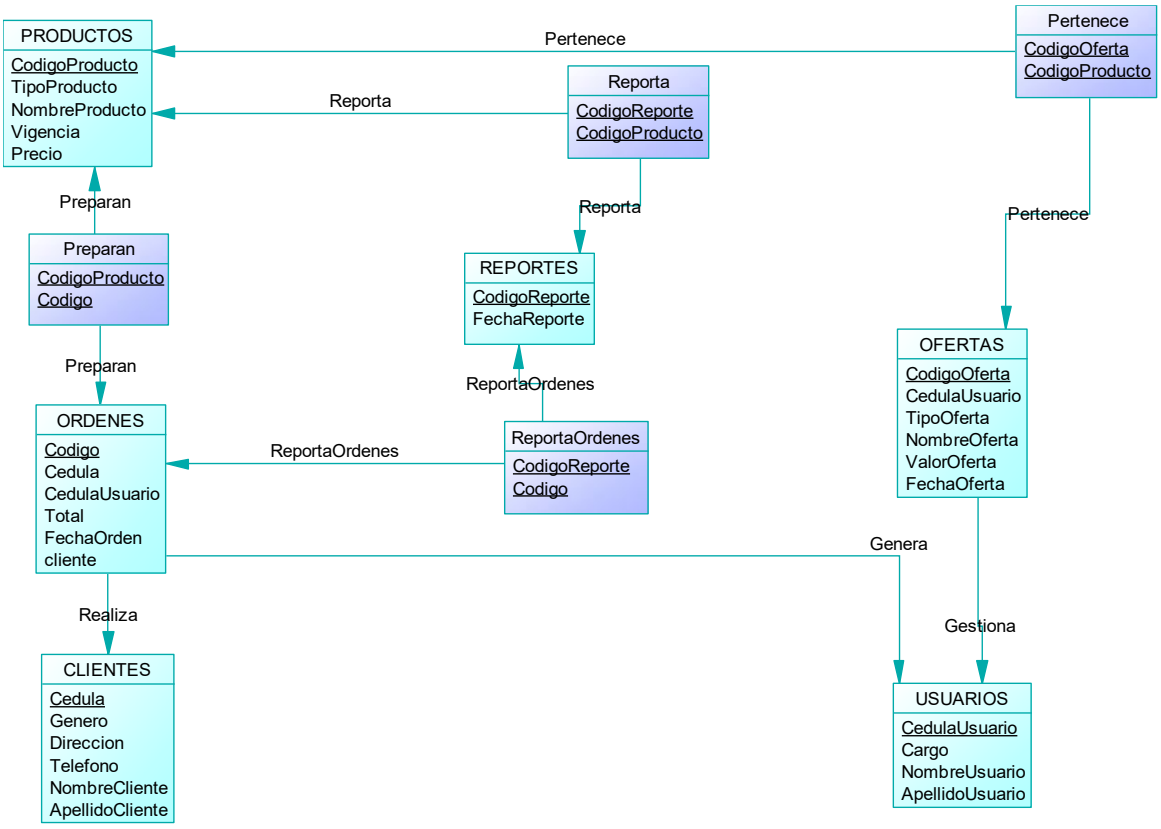


Diagrama 16: Diagrama físico de base datos

Diagramas de Secuencia

El diagrama de secuencia nos proporciona información de cómo se transmiten mensajes entre las vistas, clases, componentes del sistema o actores, así podemos controlar el manejo de mensajes, procesos y revisar que se cumplan los procedimientos de acuerdo con lo diseñado.

A continuación, se procede a presentar los respectivos diagramas de secuencia de los casos de uso antes ya presentados, el resto de diagramas de secuencia se los presentara dentro del CD entregable.

Ingreso Productos

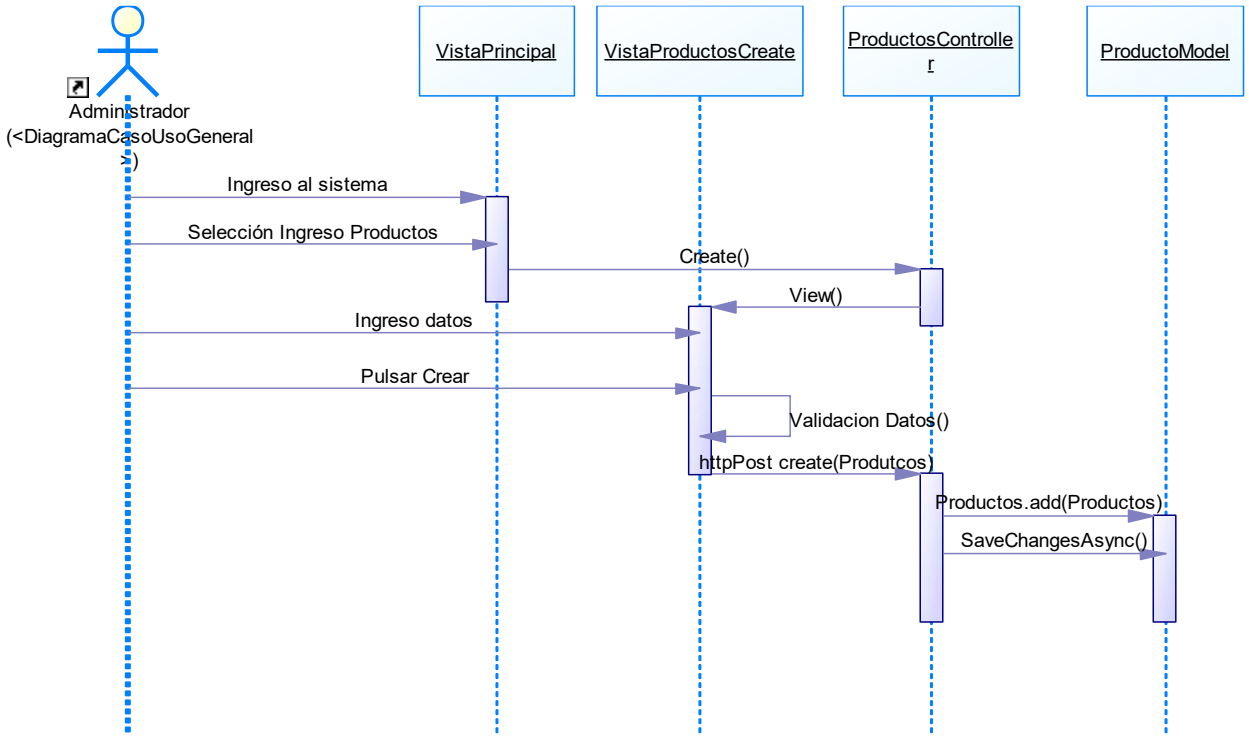


Diagrama 17: Diagrama secuencia Ingreso Productos

Modificar Productos

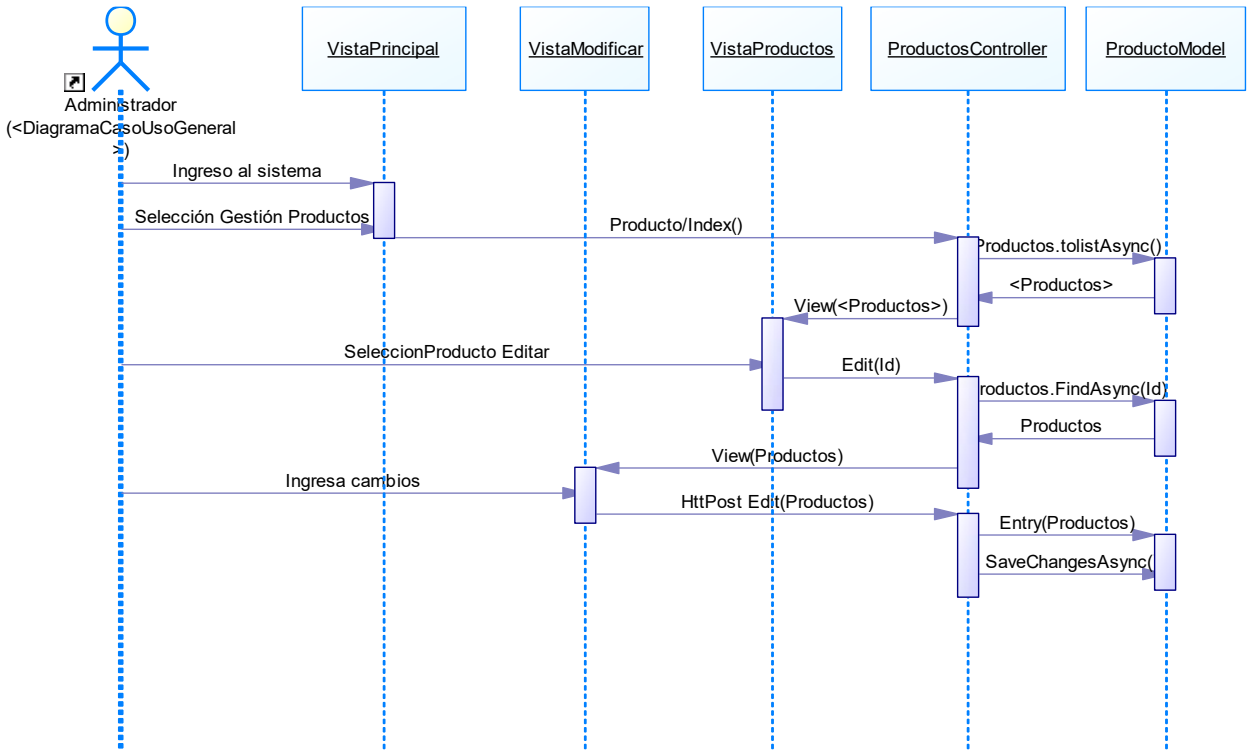


Diagrama 18: Diagrama secuencia Modificar Productos

Eliminar Productos

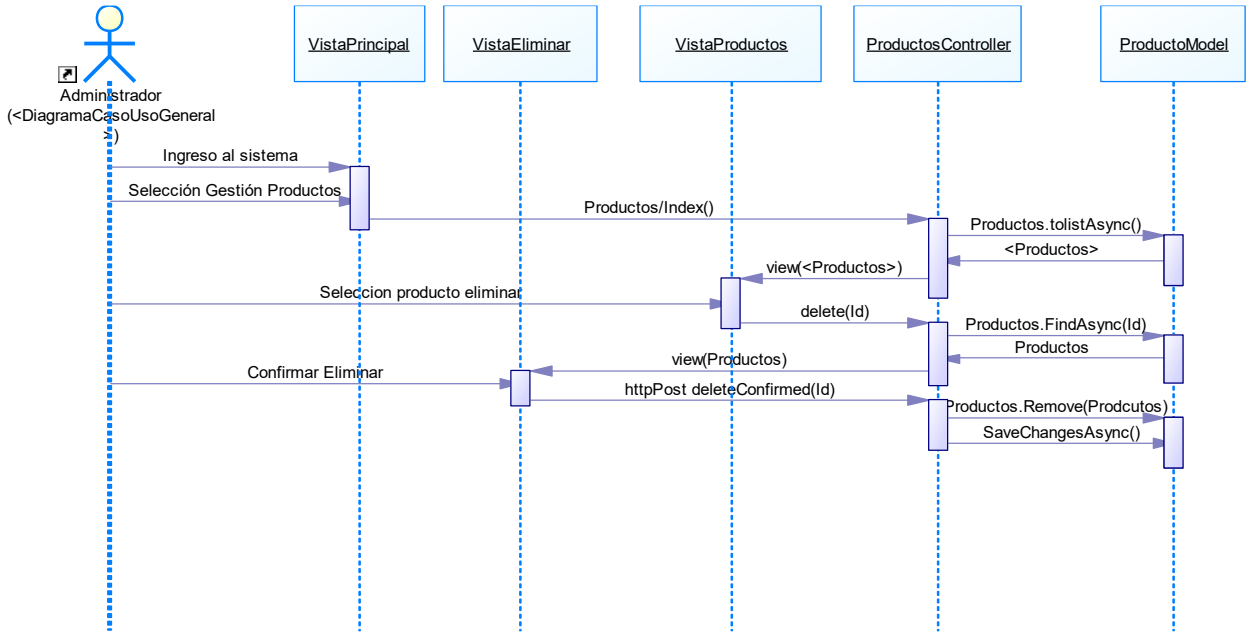


Diagrama 19: Diagrama secuencia Modificar Productos

Búsqueda Productos

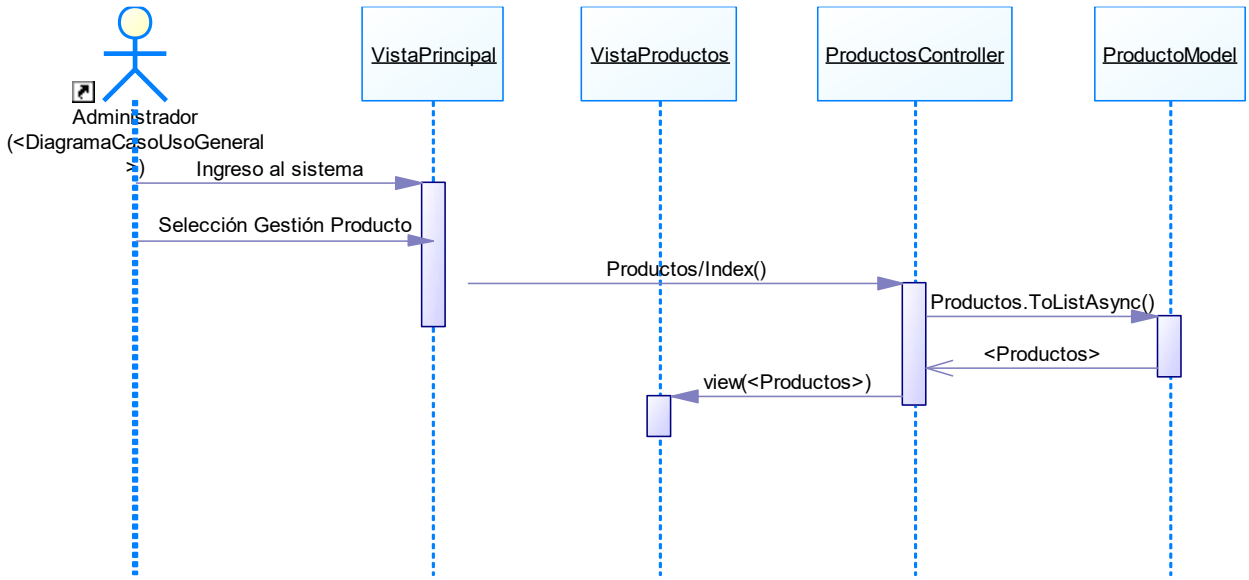


Diagrama 20: Diagrama secuencia Búsqueda Productos

Búsqueda Parámetros Productos

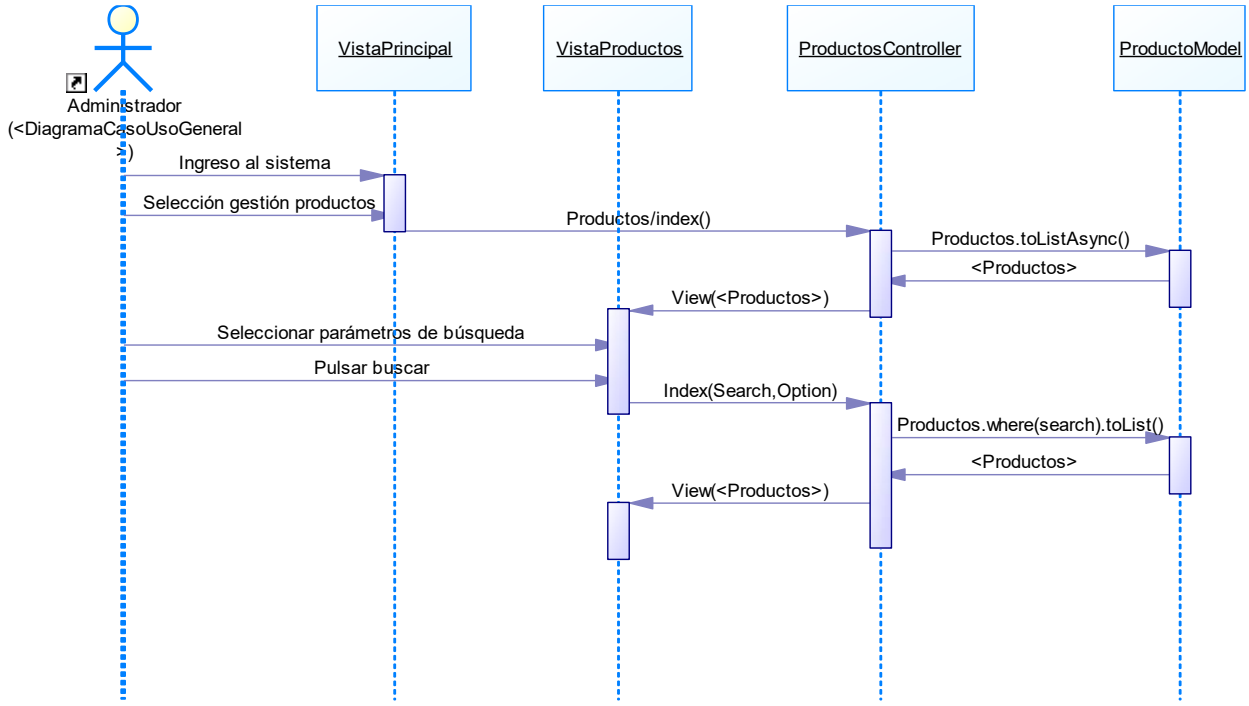


Diagrama 21: Diagrama secuencia Búsqueda Parámetros Productos

Insertar Órdenes

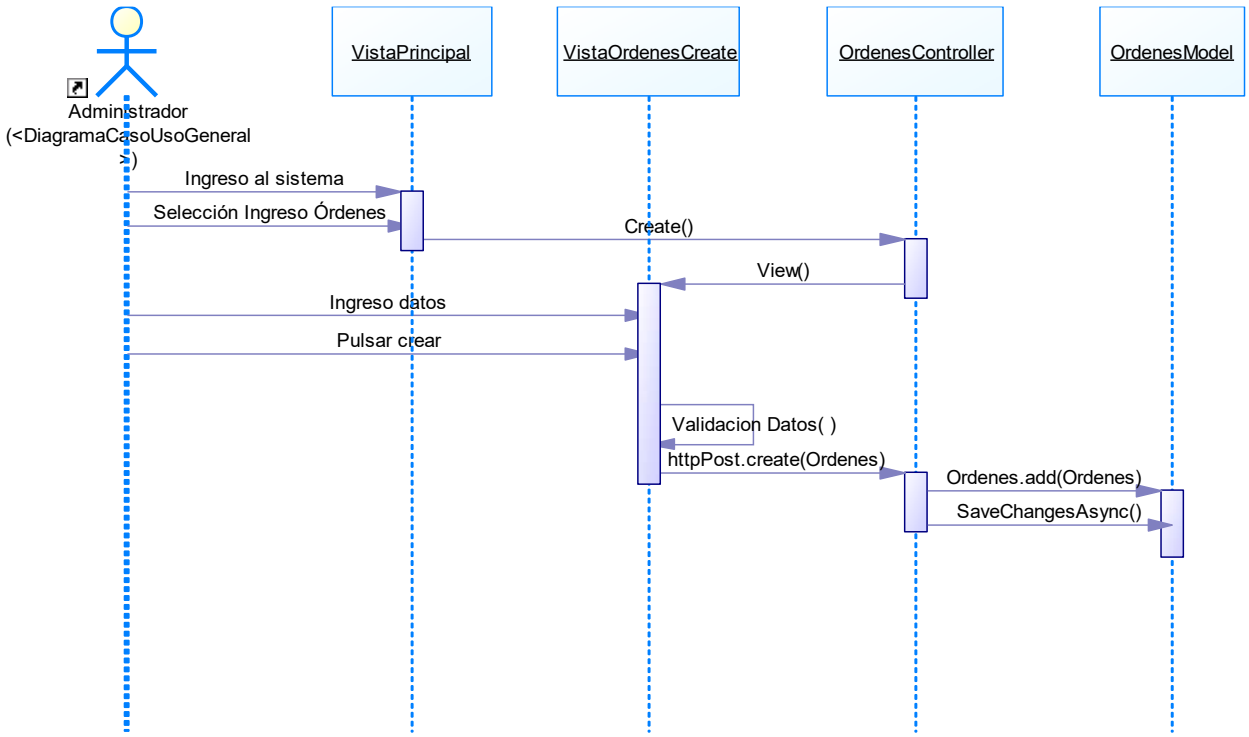


Diagrama 22: Diagrama secuencia Insertar Órdenes

Modificar Órdenes

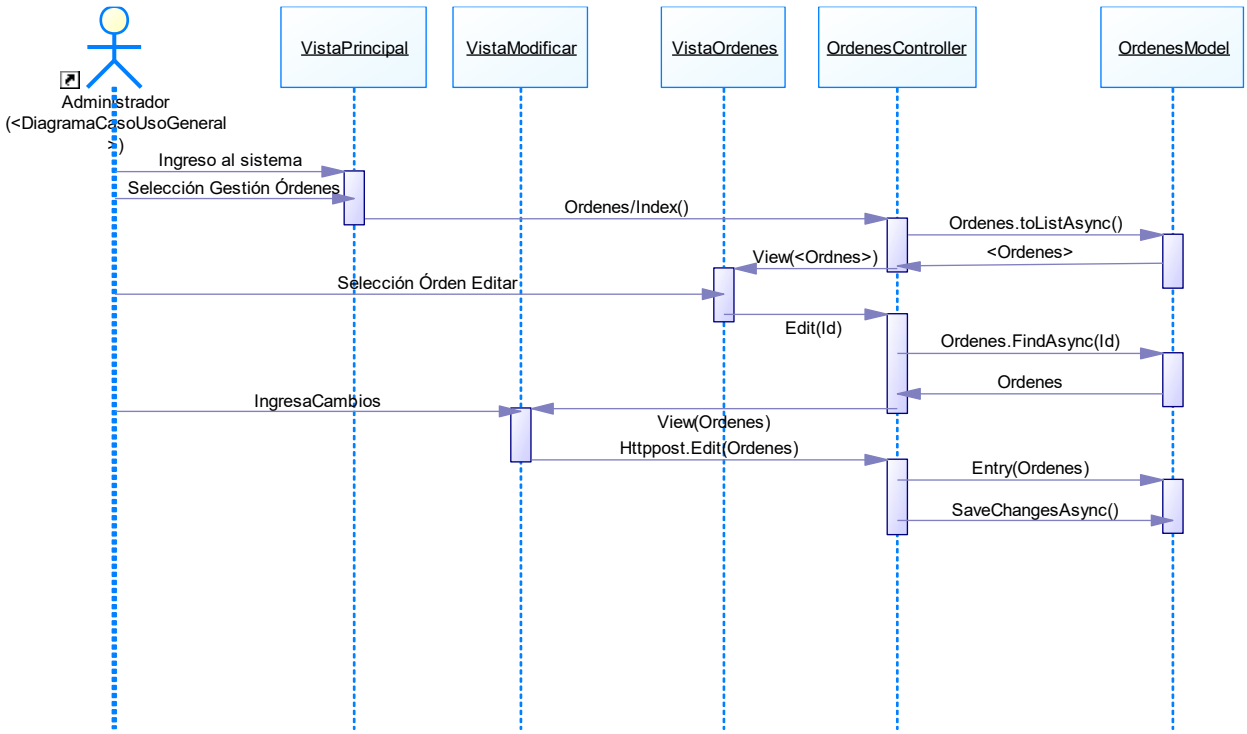


Diagrama 23: Diagrama secuencia Modificar Órdenes

Eliminar Órdenes

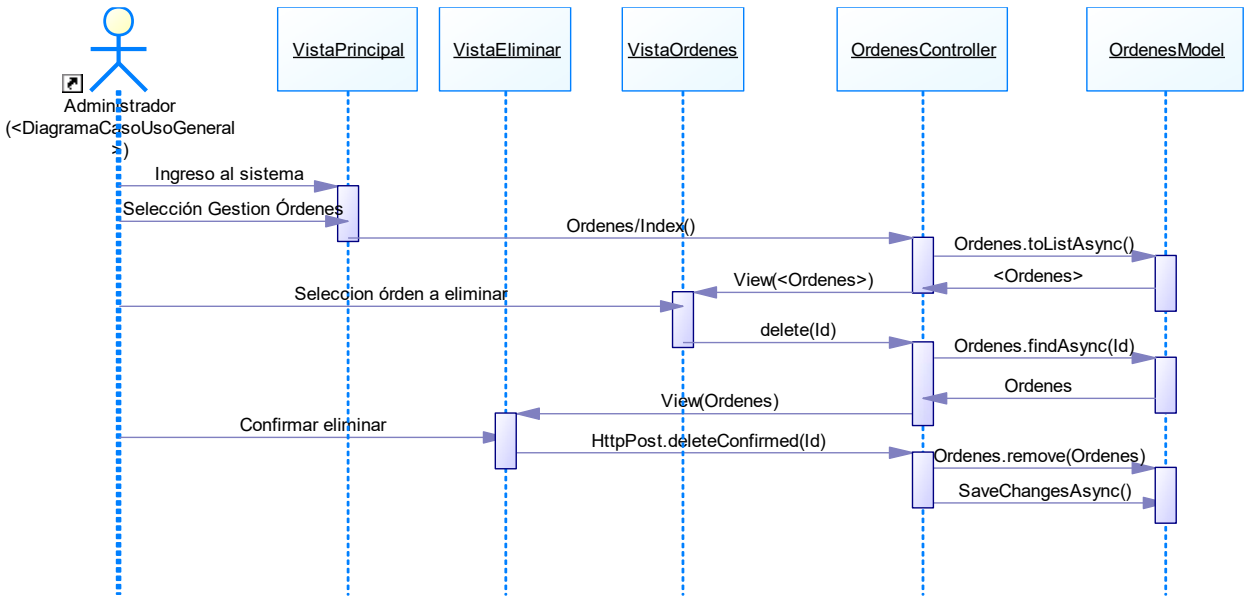


Diagrama 24: Diagrama secuencia Eliminar Órdenes

Búsqueda Órdenes

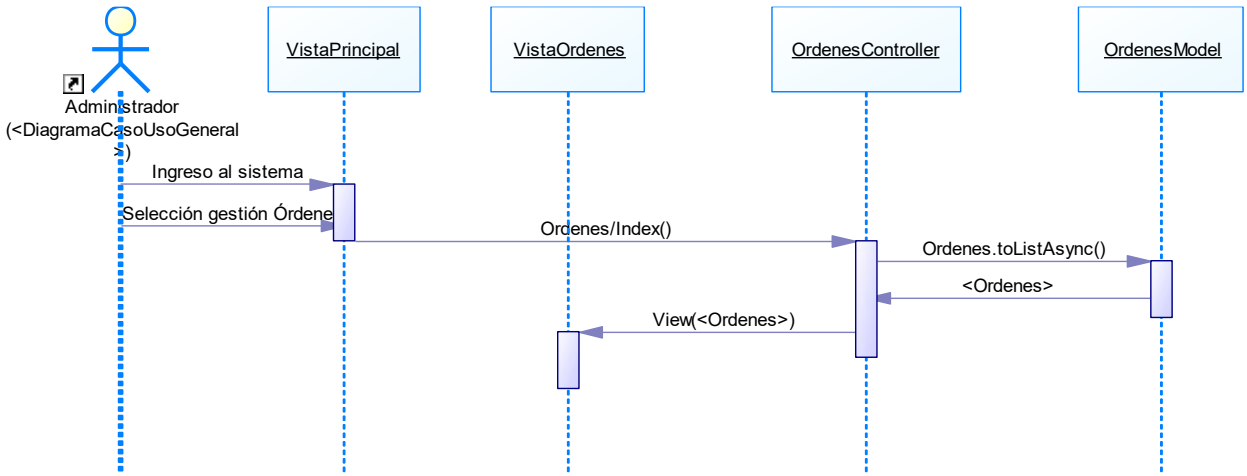


Diagrama 25: Diagrama secuencia Búsqueda Órdenes

Búsqueda Parámetros Órdenes

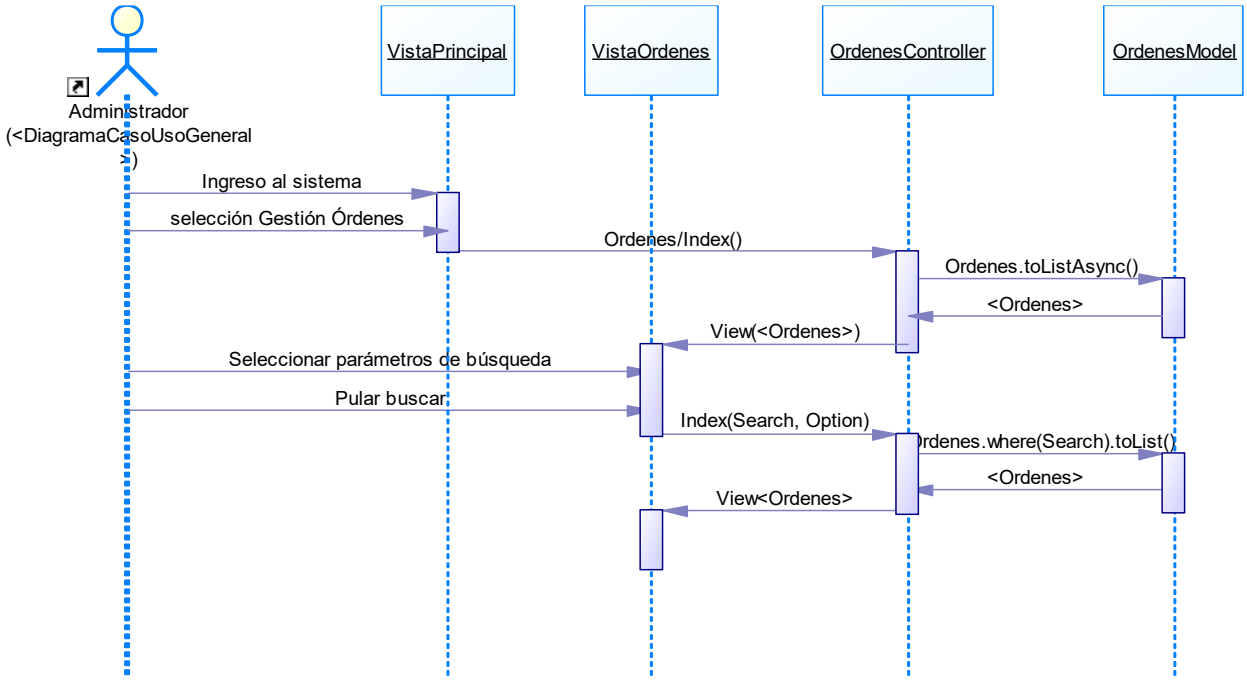


Diagrama 26: Diagrama secuencia Búsqueda parámetros Órdenes

Implementación y prueba.

Implementación

Código

Dentro del documento solamente se resaltará las partes más relevantes del código, para el resto de código consultar en el CD la carpeta Implementación.

A continuación, presentaremos fragmentos relevantes del código de Productos y Ordenes.

A continuación, en la ilustración 6, se presentan líneas de código de productos en los cuales se aprecian Index () que inicializa la vista de todos los productos y Details (string id) que nos muestran los detalles de un producto en particular.

```
// GET: Productos
public async Task<ActionResult> Index()
{
    return View(await db.PRODUCTOS.ToListAsync());
}

// GET: Productos/Details/5
public async Task<ActionResult> Details(string id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    PRODUCTOS pRODUCTOS = await db.PRODUCTOS.FindAsync(Int32.Parse(id));
    if (pRODUCTOS == null)
    {
        return HttpNotFound();
    }
    return View(pRODUCTOS);
}
```

Ilustración 6: Código Index () y Details (string id) de Producto

En la ilustración 7 podemos apreciar el código que nos indica el método de Post cuando se confirma que se desea Borrar, enviando de parámetro el id del producto.

```
// POST: ORDENES/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<ActionResult> DeleteConfirmed(string id)
{
    ORDENES oRDENES = await db.ORDENES.FindAsync(Int32.Parse(id));
    db.ORDENES.Remove(oRDENES);
    await db.SaveChangesAsync();
    return RedirectToAction("Index");
}
```

Ilustración 7: Código HttppPost DeleteConfirmed (string id)

Interfaz gráfica

Dentro del documento solamente se resaltarán las pantallas más comunes del software, el resto de la aplicación se encuentra disponible en el CD

A continuación, en la ilustración 8, se presentan pantallas del sistema en el área de productos que manejara el administrador comenzando por la pantalla que lista todos los productos.



Ilustración 8: Pantalla Index Productos

Ilustración 9 nos presenta la pantalla para edición de los campos de producto.

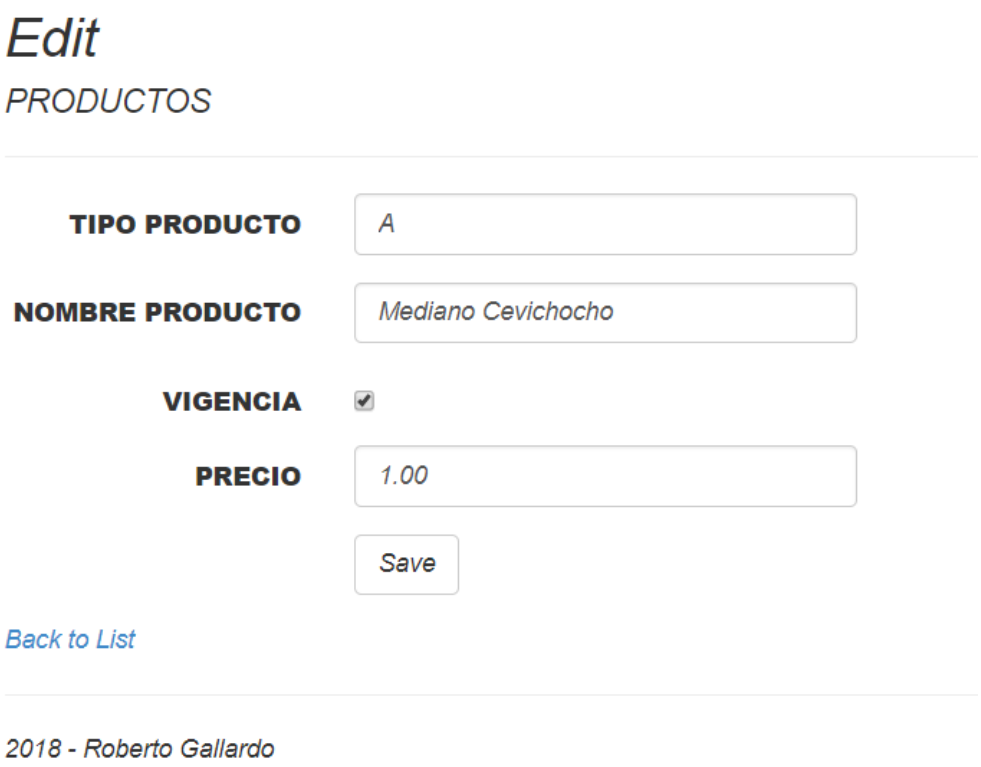


Ilustración 9: Pantalla Editar Productos

A continuación, presentaremos la pantalla de detalles del producto seleccionado en la ilustración 10:



Ilustración 10: Pantalla Detalles Producto

Pantalla para confirmar eliminación de un producto en la ilustración 11:

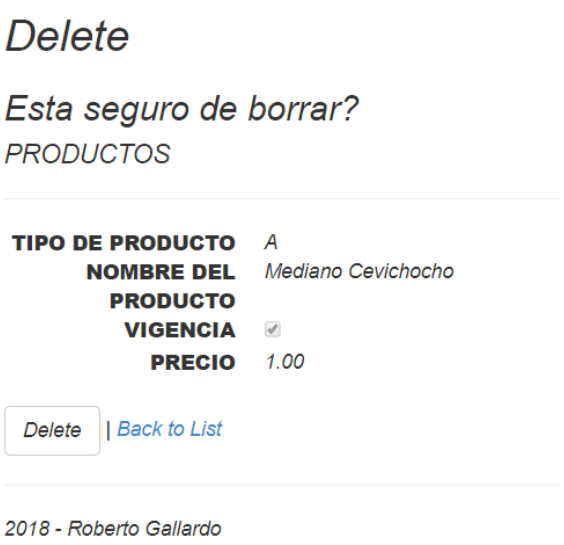


Ilustración 11: Pantalla Confirmación de eliminación producto

Pantalla Index de Órdenes listando todos los elementos existentes en la ilustración 12:

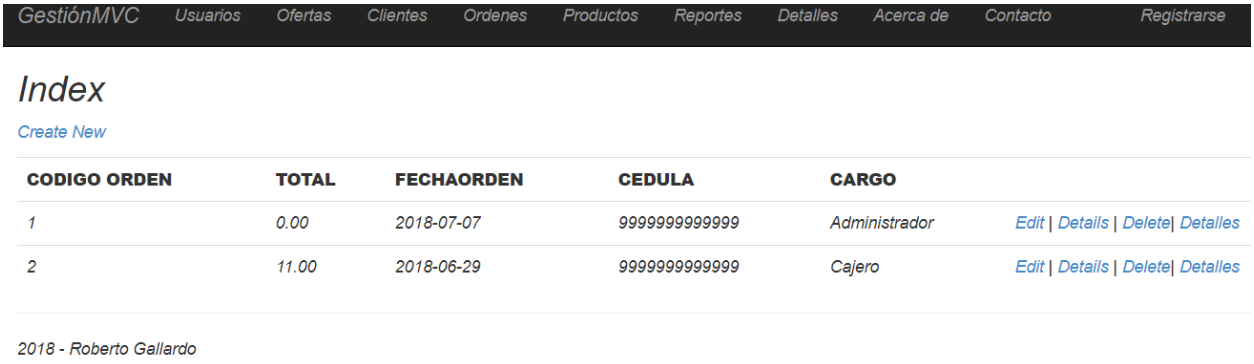


Ilustración 12: Pantalla index ordenes

Pantalla Detalles de Orden seleccionada indicando productos de la orden en la ilustración 13:



Ilustración 13: Pantalla detalles de órdenes

Pruebas

El objetivo de establecer parámetros para probar el sistema es estar consciente de que se están cumpliendo o desarrollando todos los requisitos que se diseñaron, en pocas palabras: que se compare que haga lo que indica el sistema.

A través de una lista establecemos todas las funcionalidades y requerimientos del sistema, y procedemos a ejecutar, después de una revisión si algunos requerimientos no fueron cumplidos se hará otra ronda de pruebas para comprobar que se cumplan los requisitos del sistema. (Sommerville, 2011)

Productos:

Funcionalidad	Resultado	¿Cumple?
F1.1 Insertar producto	Se ingresan productos	OK
F1.2 Modificar producto	Se modifican productos	OK
F1.3 Eliminar producto	Se eliminan productos	OK

F1.4.1 Búsqueda general producto	Se despliegan productos	OK
F1.4.2 Búsqueda por parámetro producto	Se despliegan productos por parámetro	OK

Ordenes:

Funcionalidad	Resultado	¿Cumple?
F2.1 Insertar orden	Se ingresan orden	OK
F2.2 Modificar orden	Se modifican orden	OK
F2.3 Eliminar orden	Se eliminan orden	OK
F2.4.1 Búsqueda general orden	Se despliegan ordenes	OK
F2.4.2 Búsqueda por parámetro orden	Se despliegan orden por parámetro	OK

Reportes:

Funcionalidad	Resultado	¿Cumple?
F3.1.1 Generar reporte Cierre Caja	Se ingresan reportes cierre caja	NO
F3.1.2 Generar reporte producto Estrella	Se ingresan reportes producto estrella	NO

Usuarios:

Funcionalidad	Resultado	¿Cumple?
F4.1 Insertar Usuarios	Se ingresan Usuarios	OK
F4.2 Modificar Usuarios	Se modifican Usuarios	OK
F4.3 Eliminar Usuarios	Se eliminan Usuarios	OK
F4.4.1 Búsqueda general Usuarios	Se despliegan Usuarios	OK
F4.4.2 Búsqueda por parámetro Usuarios	Se despliegan Usuarios por parámetro	OK

En la sección reportes no hubo resultados esperados, lo cual nos lleva a una segunda etapa de pruebas mientras se corrige el problema.

Reportes (Segunda Prueba):

Funcionalidad	Resultado	¿Cumple?
F3.1.1 Generar reporte Cierre Caja	Se ingresan reportes cierre caja	OK
F3.1.2 Generar reporte producto Estrella	Se ingresan reportes producto estrella	OK

Implantación.

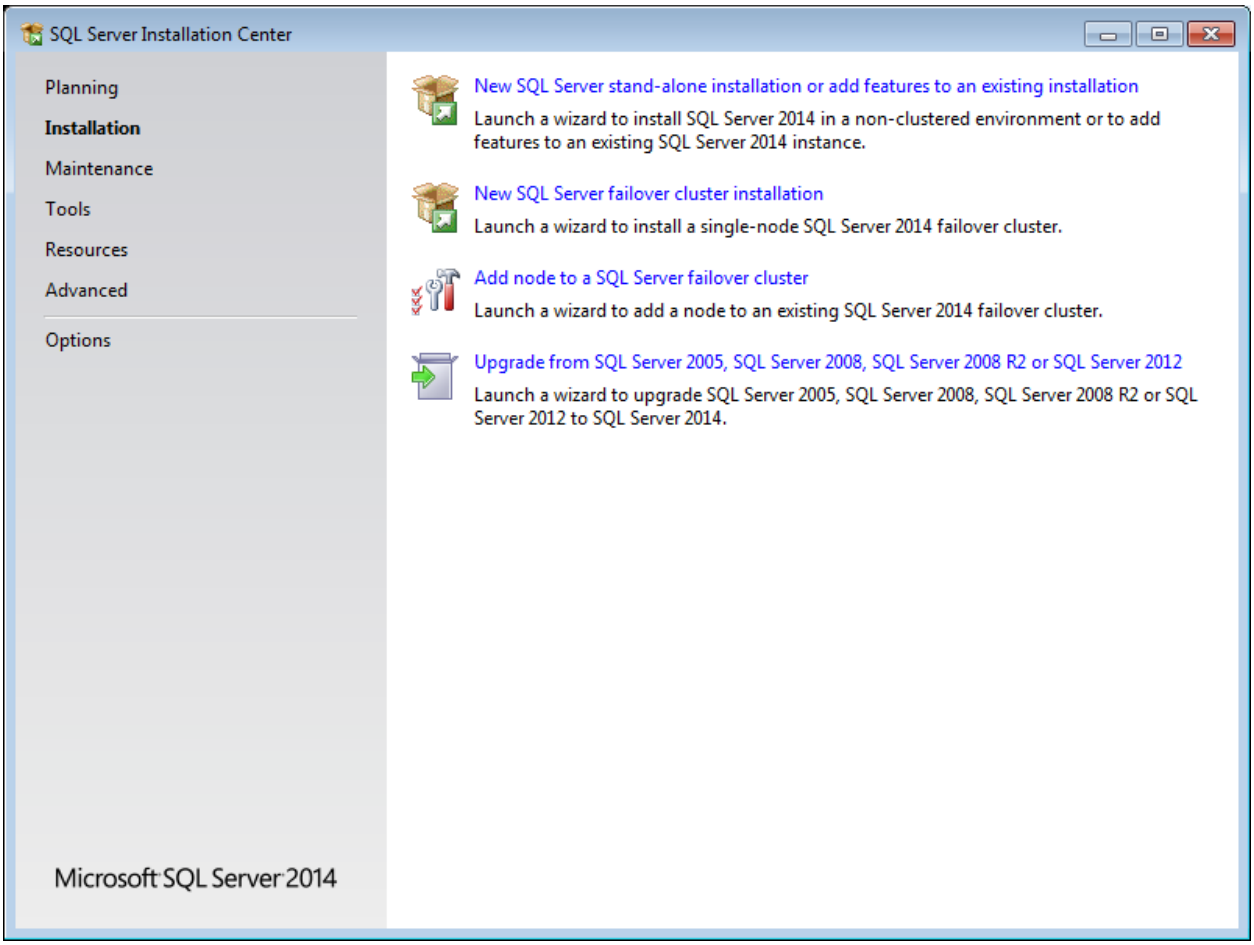
Una vez que el cliente dio por aceptada las funcionalidades que brinda la aplicación, se procedió a implantar dentro del restaurante, para lo cual se explicara a continuación el proceso:

- El equipo disponible en el cual se implantará el sistema es el siguiente, de procesador posee un núcleo i5 de Intel, memoria RAM 4Gb, es un computador de escritorio.
- Instalación de Internet Information Services(IIS), se instala el servidor web local para alojar la aplicación, como es modular al instalar un nuevo sitio web nos genera el dominio local, redirigiendo a la carpeta donde se ejecutará la aplicación en el servidor a continuación en la ilustración 14 se indican los parámetros de nombre de sitio y el acceso físico.



Ilustración 14: características del servicio local en IIS

- Instalación SQL Server 2014, la instalación del gestor de base de datos se la hace con una interfaz gráfica para comodidad del manejo de la base de datos.



Funcionamiento y mantenimiento

El funcionamiento de la aplicación se la puede establecer en dos formas diferentes de acuerdo con las necesidades del cliente.

Cada forma del sistema esta detallada con un diagrama representativo en las ilustraciones 14 y 15.

Primera manera es a través de la nube(Internet), sin necesidad de ningún dispositivo aparte de el que se usará para ejecutar la aplicación.

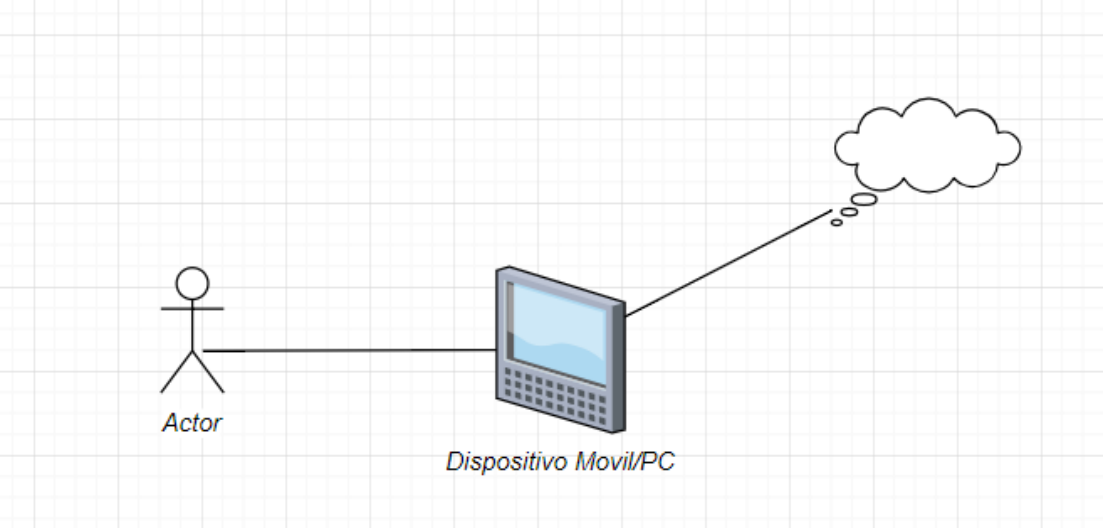


Ilustración 15: Primera opción de instalación sistema

Segunda manera de instalar el sistema es a través de un servidor local que se encuentre en el local conectado a través de la red de área local(LAN).

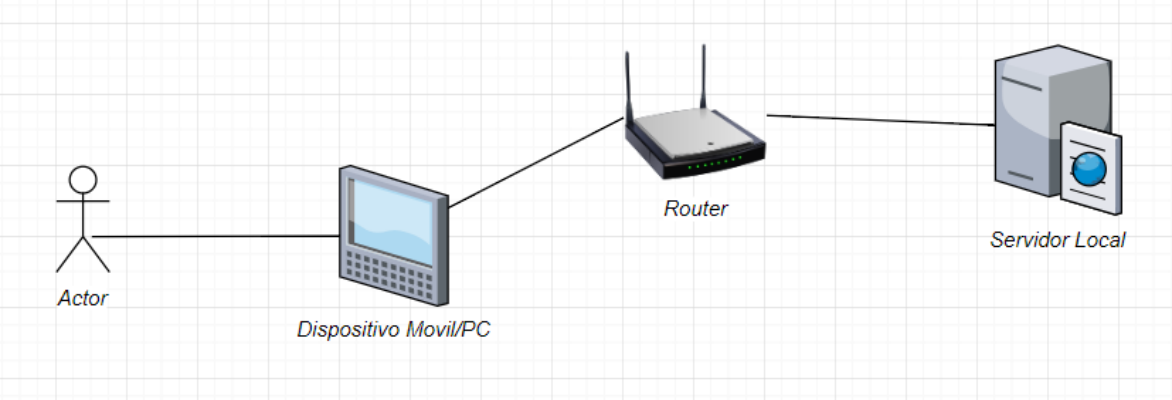


Ilustración 16 Segunda opción de instalación del sistema

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.

Conclusiones

- La presente disertación se dedicó al desarrollo de una aplicación web que pretende facilitar el manejo de las órdenes de comida dentro del restaurante CHOCHOTE, logrando así un proceso más controlado y lograr establecer mejor la organización al momento de servir las órdenes a sus clientes.
- A lo largo de la elaboración del sistema se encontraron varias incógnitas por resolver, las cuales fueron resueltas con constante análisis de los requerimientos del cliente, como por ejemplo tener en cuenta que el diseño esté alineado con las expectativas de los requerimientos del usuario final.
- La comunicación con el cliente es crucial a la hora de obtener requerimientos junto con documentación del proceso ya que, si el cliente no entiende lo que necesita, ese requerimiento puede ser mal interpretado, llevándonos así al mal funcionamiento del sistema.
- Las pruebas del sistema son cruciales para verificar el correcto y total funcionamiento de la aplicación, logrando también comprobar si cumple lo requerimientos necesarios.
- El método de cascada resulto muy útil en el desarrollo del proceso ya que no hay variaciones en el proceso y no se requiere la presencia constante del cliente, ahorrándonos tiempo en el desarrollo del proyecto.
- El patrón de diseño MVC nos establece especial atención en cómo está organizada la aplicación, separándola en tres capas física y lógicamente.

Recomendaciones

- El desarrollo orientado a web nos brinda una gran movilidad ya que el único requisito importante es que se tenga un navegador web, ayudándonos así a gestionar las ordenes sin necesidad de tener un puesto fijo.
- Se recomienda la migración a la nube para no tener la necesidad de dar mantenimiento a los servidores y poder establecer más accesos en caso de abrir nuevas sucursales.
- Dar mantenimiento al servidor actual, actualizando hardware para un mejor rendimiento del sistema y optimizar tiempos de procesamiento.

CAPÍTULO V: BIBLIOGRAFÍA.

Alegria, I. R. (2012). *Informatica aplicada.* Retrieved from https://irfeyal.wordpress.com/bases-de-datos/modelamiento-de-bdd/

Francia, J. (2017, Septiembre 25). *Scrum.org.* Retrieved from https://www.scrum.org/resources/blog/que-es-scrum

Gómez, V. (n.d.). *Arquitectura en Tres Capas*. Retrieved from https://instintobinario.com/arquitectura-en-tres-capas/

Houston, U. o. (2002, Mayo). *University of Houston.* Retrieved from http://sce.uhcl.edu/helm/rationalunifiedprocess/index.htm

Machala, U. t. (2010, Noviembre 21). *Estrategias metodologicas*. Retrieved from http://estrategiasmetodologicasinformaticas.blogspot.com/2011/01/metodologia-informatica.html

Microsoft. (2009, Abril 14). *Microsoft Developer.* Retrieved from https://blogs.msdn.microsoft.com/ricom/2010/04/14/my-history-of-visual-studio-epilog/

Officer, O. o. (2009, Octubre). *AN INTRODUCTION TO OBJECT ORIENTED METHODOLOGY (OOM)* . Retrieved from https://www.ogcio.gov.hk/en/infrastructure/methodology/system_development/past_documents/oom/doc/g52a_pub.pdf

Powell-Morse, A. (2017, Marzo 14). *Rational Unified Process*. Retrieved from https://airbrake.io/blog/sdlc/rational-unified-process

Radigan, D. (2017). *Scrum*. Retrieved from https://es.atlassian.com/agile/scrum

SAP. (2018, 01 15). *SAP.* Retrieved from https://www.sap.com/products/powerdesigner-data-modeling-tools.html

Sommerville, I. (2011). *Ingenieria del Software, Septima edición.* Madrid: Pearson. Retrieved from http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf

Tinoco Gómez, O., Rosales López, P. P., & Bacalla, S. (2010, Julio). *redalyc*. Retrieved from http://www.redalyc.org/comocitar.oa?id=81619984009

Turner, D. N. (2015). *Research Methodology. The Origin and Scope of Thick Ethnography.* Retrieved from https://www.grin.com/document/306758

Valladolid, D. d. (2009, 01 29). *Ingeniería del Software. Tema 4*. Retrieved from https://www.infor.uva.es/~jvalvarez/docencia/ptema4is1.pdf