

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS



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

Levantamiento de los principales procesos para el Restaurante y Servicio de Cáterin Alexander; y automatización del proceso de inventario y el proceso de gestión de reserva de mesas mediante una aplicación basada en Java aplicando la metodología de Programación Rational Unified Process (RUP)

DAVID MORÁN

JORGE CUNALATA

DIRECTOR: OSWALDO ESPINOSA

QUITO, 2016

Resumen

La automatización de procesos dentro de la industria ha crecido en las últimas décadas significativamente debido al crecimiento tecnológico y a la importancia de tener procesos bien definidos. Esto ha impulsado a varias empresas a realizar un levantamiento de procesos y automatización de los mismos para llevar un mayor control.

El Restaurante y Servicio de Cáterin Alexander no constaba con procesos definidos es por esto que hemos visto la importancia de realizar del levantamiento de los principales procesos y la automatización del proceso de inventario y el proceso de gestión de reserva para que puedan llevar así un mejor control de su negocio.

Dedicatoria

Dedicamos esta disertación a nuestras familias que han sido un apoyo fundamental en toda nuestra carrera. Agradecemos también a nuestros compañeros y profesores que fueron pilares fundamentales en nuestro crecimiento.

ÍNDICE

Resumen.....	2
Dedicatoria.....	3
CAPÍTULO 1	9
1. INTRODUCCIÓN	9
1.1 Metodología de Programación Rational Unified Process (RUP)	9
1.1.1 Introducción	9
1.1.2 Características Fundamentales.....	9
1.1.3 Fases y Ciclo de Vida	10
1.1.4 Ventajas y Desventajas.	14
1.1.5 Aplicación de RUP en el desarrollo.....	15
1.2 Lenguaje de programación Java	16
1.2.1 Introducción	16
1.2.2 Historia del lenguaje de programación Java	17
1.2.3 Características fundamentales.....	18
1.2.4 Ventajas y Desventajas	19
1.2.5 Aplicaciones de Java.....	20
1.3 Base de datos PostgreSQL	20
1.3.1 Introducción	20
1.3.2 Breve historia de PostgreSQL.....	21
1.3.3 Características fundamentales.....	21
1.3.4 Ventajas y desventajas	22
1.3.5 Aplicaciones de PostgreSQL	23
CAPÍTULO 2	24
2. FASE DE INICIO	24
2.1 Levantamiento de los principales procesos	24
2.1.1 Cadena de valor.....	24
2.1.2 Identificación principales procesos.....	24
2.1.3 Diagramación y descripción del proceso de inventarios.....	25
2.1.4 Diagramación y descripción del proceso de gestión de reserva de mesas	27

2.2	Modelado del negocio	29
2.2.1	Análisis de situación actual del negocio	29
2.2.2	Descripción breve del sistema	30
2.2.3	Descripción breve de las necesidades del negocio en el sistema propuesto	30
2.2.4	Beneficios y mejoras que producirá el sistema.....	31
2.3	Requerimientos	32
2.3.1	Descripción general del sistema.....	32
2.3.2	Requerimientos funcionales.....	33
2.3.3	Requerimientos de hardware y software.....	33
2.3.4	Diagrama de Casos de Uso	34
CAPÍTULO 3.	47
3.	FASE DE ELABORACIÓN	47
3.1	Análisis y diseño	47
3.1.1	Análisis del sistema actual	47
3.1.2	Estructura del Producto.....	48
3.1.3	Diseño de Interfaces.....	49
3.1.4	Diagrama de Clases.....	54
3.1.5	Diagrama de Secuencia.....	56
3.1.6	Diagrama Entidad Relación	74
3.1.7	Diagrama Entidad Relación	76
CAPÍTULO 4	77
4.	FASE DE CONSTRUCCIÓN	77
4.1	Implementación	77
4.1.1	Estándares de implementación.....	77
4.1.2	Definición de clases	78
4.1.3	Codificación de Clases.....	79
4.1.4	Conexión con Base de Datos	79
4.1.5	Manual Técnico	79
4.1.6	Manual de Usuario.....	79
4.1.7	Código Fuente.....	79
4.2	Pruebas	79
4.2.1	Plan de pruebas del sistema	79

4.2.2	Informe de los resultados de las pruebas	84
5.	FASE DE TRANSICIÓN	88
5.1	Despliegue.....	88
5.1.1	Software empaquetado.....	88
5.1.2	Instalación del software	92
6.	CONCLUSIONES Y RECOMENDACIONES.....	95
6.1	Conclusiones	95
6.2	Recomendaciones	96
Bibliografía	97
Glosario	98

ÍNDICE DE ILUSTRACIONES

Ilustración 1:	Fases del ciclo de RUP	11
Ilustración 2:	La vida de un proceso consta de ciclos de vida desde su nacimiento hasta su muerte	11
Ilustración 3:	RUP con sus flujos de trabajo principales y de soporte y control	12
Ilustración 4:	Fases de prueba en el proceso de creación o evolución de un software	14
Ilustración 5:	Evolución de los lenguajes de programación	16
Ilustración 6:	Cadena de valor del Restaurante y Servicios de Cáterin Alexander	24
Ilustración 7:	Procesos principales y subprocesos del Restaurante y Servicios de Cáterin Alexander.....	24
Ilustración 8:	Diagrama del proceso de Inventario	25
Ilustración 9:	Diagrama del proceso de Inventario	27
Ilustración 10:	Diagrama de Casos de Uso General	34
Ilustración 11:	Diagrama de Casos de Uso Gestión de Reserva de Mesas	35
Ilustración 12:	Diagrama de Casos de Uso Crear Una Reserva de Mesa	35
Ilustración 13:	Diagrama de Casos de Uso Modificar Una Reserva de Mesa	37
Ilustración 14:	Diagrama de Casos de Uso Eliminar Una Reserva de Mesa	38
Ilustración 15Gráfico 14:	Diagrama de Casos de Uso Consultar Una Reserva de Mesa	39
Ilustración 16:	Diagrama de Casos de Uso Gestión de Inventario	41
Ilustración 17:	Diagrama de Casos de Uso Crear Inventario.....	41
Ilustración 18:	Diagrama de Casos de Uso Eliminar Inventario.....	42
Ilustración 19:	Diagrama de Casos de Uso Gestión de Inventario	44
Ilustración 20:	Diagrama de Casos de Uso Crear Inventario.....	44
Ilustración 21:	Diagrama de Casos de Uso Eliminar Inventario.....	45

Ilustración 22: Programacion en Capas	48
Ilustración 23: Interfaz Menú Principal	49
Ilustración 24: Interfaz Proveedores	50
Ilustración 25: Interfaz Productos	51
Ilustración 26: Interfaz Inventario	52
Ilustración 27: Interfaz Reserva de Mesas	53
Ilustración 28: Diagrama de Clases Reserva de Mesas	54
Ilustración 29: Diagrama de Clases Proceso de Inventario	55
Ilustración 30: Diagrama de Secuencia Insertar Proveedor	56
Ilustración 31: Diagrama de Secuencia Eliminar Proveedor	57
Ilustración 32: Diagrama de Secuencia Consultar Proveedor.....	58
Ilustración 33: Diagrama de Secuencia Modificar Proveedor	59
Ilustración 34: Diagrama de Secuencia Insertar Producto	60
Ilustración 35: Diagrama de Secuencia Eliminar Producto	61
Ilustración 36: Diagrama de Secuencia Consultar Producto.....	62
Ilustración 37: Diagrama de Secuencia Modificar Producto	63
Ilustración 38: Diagrama de Secuencia Crear Inventario	64
Ilustración 39: Diagrama de Secuencia Eliminar Inventario	65
Ilustración 40: Diagrama de Secuencia Crear Reporte de Inventario.....	66
Ilustración 41: Diagrama de Secuencia Crear Cliente	67
Ilustración 42: Diagrama de Secuencia Eliminar Cliente	68
Ilustración 43: Diagrama de Secuencia Modificar Cliente.....	69
Ilustración 44: Diagrama de Secuencia Crear Reserva de Mesas	70
Ilustración 45: Diagrama de Secuencia Eliminar Reserva de Mesas.....	71
Ilustración 46: Diagrama de Secuencia Consultar Reserva de Mesas	72
Ilustración 47: Diagrama de Secuencia Modificar Reserva de Mesas.....	73
Ilustración 48: Diagrama Entidad Relación Reserva de Mesas	74
Ilustración 49: Diagrama Entidad Relación Proceso de Inventario	75
Ilustración 50: Empaquetado del Programa y Generación del Ejecutable.....	89
Ilustración 51: Empaquetado del Programa y Generación del Ejecutable.....	90
Ilustración 52: Empaquetado del Programa y Generación del Ejecutable.....	91
Ilustración 53: Empaquetado del Programa y Generación del Ejecutable.....	92
Ilustración 54: Instalación del Software RestauranteAlexander	93
Ilustración 55: Instalación del Software RestauranteAlexander	93
Ilustración 56: Instalación del Software RestauranteAlexander	94

ÍNDICE DE TABLAS

Tabla 1: Diccionario de Datos	76
Tabla 2: Plan de Pruebas del Sistema – Gestión de Reserva de Mesas	80
Tabla 3: Plan de Pruebas del Sistema – Gestión de Proceso de Inventario	82
Tabla 4: Resultado de Pruebas del Sistema – Gestión de Reserva de Mesas	84
Tabla 5: Resultado de Pruebas del Sistema – Gestión Proceso de Inventario	86
Tabla 6: Resultado de Pruebas del Sistema – Menú Principal.....	87

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Metodología de programación Rational Unified Process (RUP)

1.1.1 Introducción

El Proceso Unificado de Rational es una metodología de desarrollo de software^[a] iterativa, estructurada y adaptable al contexto y necesidades de cada organización. El desarrollo de software son un conjunto de actividades que se realizan para transformar las necesidades o requerimientos de las organizaciones en sistemas funcionales que satisfacen dichas necesidades.

La empresa Rational Software fue quien desarrolló el Proceso Unificado de Rational que actualmente es propiedad de IBM^[b] y es una de las metodologías^[c] más utilizada al momento de realizar análisis, diseño, implementación y documentación de sistemas orientados a objetos.

La metodología RUP trabaja conjuntamente con el Lenguaje de Modelado Unificado (UML) ya que se enfoca a un modelado visual del software que se está desarrollando.

1.1.2 Características fundamentales

Las principales características que podemos resaltar del Proceso Unificado (RUP) son las siguientes:

^[a] Software: Es un conjunto de instrucciones que los procesadores emplean para manipular datos.

^[b] IBM: International Business Machines empresa dedicada al desarrollo de software y hardware.

^[c] Metodología: Etapa dentro de un proyecto en la que se guía mediante métodos y teoría para la realización de ciertas tareas.

- Adaptabilidad a cualquier tipo de proyecto gracias a que implementa las mejores prácticas de ingeniería de software.
- En cada fase se presentan entregables (artefactos)^[d].
- Se guía con UML^[e] (Unified Modeling Language) casos de uso.
- En caso de cambios se maneja con Control de Cambios.
- Es un desarrollo iterativo e incremental.
- Verificación de calidad del software.
- Se manejan roles dentro del proceso de desarrollo.

1.1.3 Fases y ciclo de vida

El proceso unificado de Rational se desarrolla mediante ciclos.

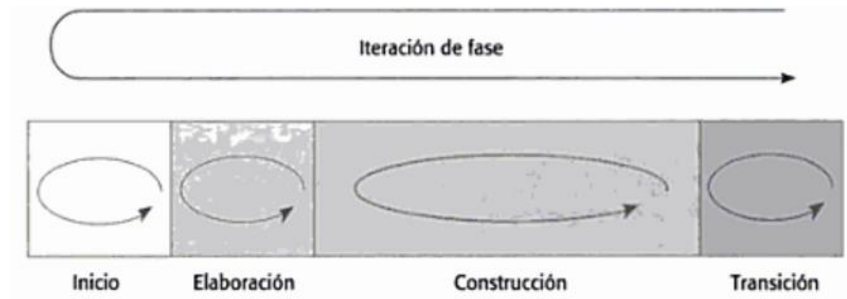
Cada ciclo comprende cuatro fases las cuales son:

- Inicio
- Elaboración
- Construcción
- Transición

^[d] Artefacto: Es un resultante tangible dentro del proceso de desarrollo de software.

^[e] UML: Lenguaje Unificado de Modelado es un estándar para crear esquemas, diagramas y documentación en el desarrollo de software.

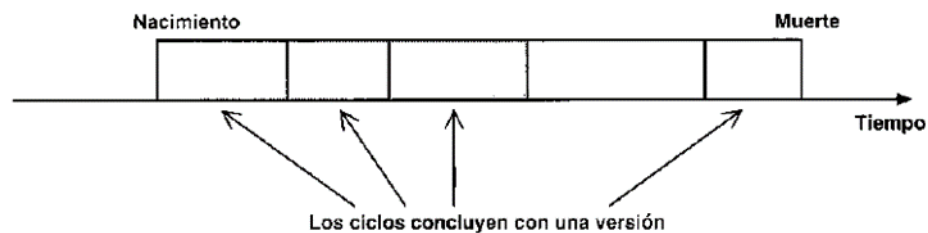
Ilustración 1: Fases del ciclo de RUP



Fuente: Ian Sommerville (2005). *Ingeniería del Software*, página 77, Séptima Edición.

Hay que tener en cuenta que con cada ciclo que se finaliza se crea un artefacto o entregable (nueva versión o mejora del software), como observamos en el siguiente gráfico:

Ilustración 2: La vida de un proceso consta de ciclos de vida desde su nacimiento hasta su muerte



Fuente: Jacobson, Booch, Rumbaugh (2000). *El Proceso Unificado de Desarrollo de Software*, página 8.

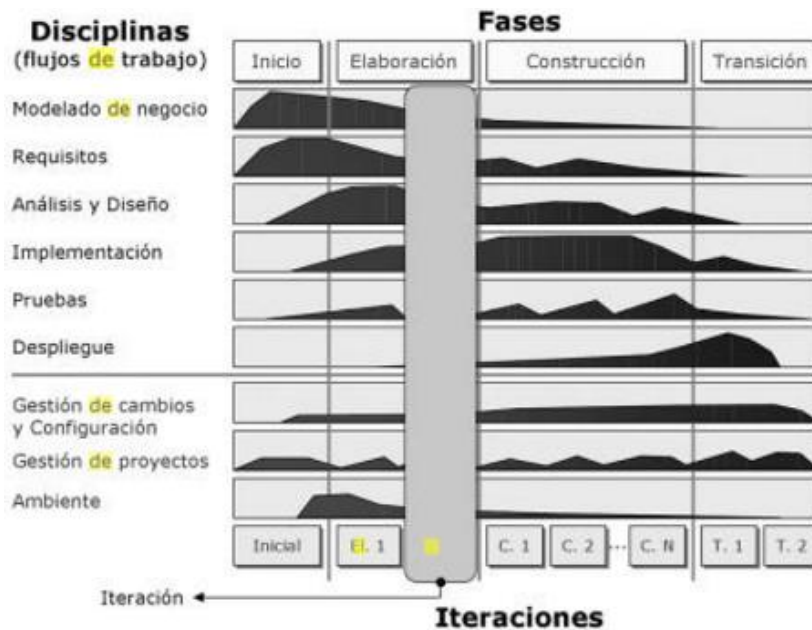
Cada ciclo puede ser subdividido o atendido en seis flujos de trabajo principales como son:

- Modelado del negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

También existen tres flujos de soporte que deben ser analizados y estos son:

- Gestión de cambios y Configuración.
- Gestión de proyectos.
- Ambiente.

Ilustración 3: RUP con sus flujos de trabajo principales y de soporte y control



Fuente: Alonso, F; Martínez, L; Segovia, J. (2005). *Introducción a la Ingeniería de Software Modelos de Desarrollo de Programas*, página 336

La fase de inicio lo que tiene por objetivo es realizar un análisis sobre el negocio, determinar su alcance, riesgos del proyecto y definir las fases en las que se desarrollará el proyecto tomando en cuenta los requisitos y necesidades de la organización.

La fase de elaboración tiene como fin definir la arquitectura con la que se desarrollará el proyecto, es decir, los casos de uso^[f] más relevantes en torno al negocio. También se

^[f] Casos de uso: Diagrama en el que se indica las transacciones que realiza un actor dentro de un Sistema.

toma en cuenta los estimados en cuanto tiempo, líneas de código y recursos monetarios durante todo el desarrollo.

La fase de construcción es donde se añade la implementación del código a la arquitectura^[g], se busca cubrir todos los requerimientos que la organización tiene, aunque es muy probable que existan defectos estos se los podrá solucionar en la fase de transición.

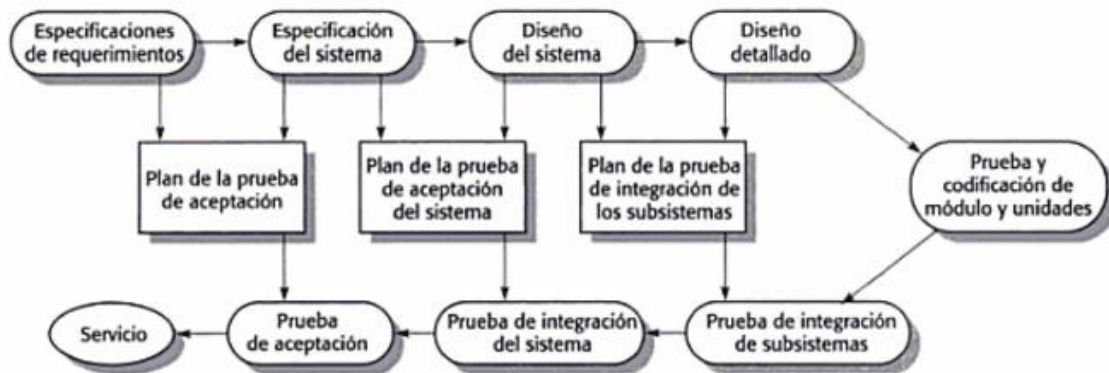
La fase de transición es la fase donde se lanza un producto beta^[h] el cual es entregado a una parte de usuarios para que lo manipulen e informen de errores los cuales pueden ser clasificados en dos, los de más impacto en el funcionamiento que exigen a los desarrolladores la mejora y un nuevo lanzamiento de un producto versión “delta”, y los errores de menor impacto que suelen ser más como sugerencias que pueden ser realizados o no en nuevas versiones futuras del producto. En esta fase también es donde se brinda capacitaciones sobre el producto, asistencia post entrega y mantenimientos.

Durante el desarrollo del proyecto tenemos etapas que sobresalen las cuales son:

^[g] Arquitectura: Es la relación entre las partes que constituyen un Sistema.

^[h] Producto beta: Es un producto lanzado a los usuarios para pruebas públicas.

Ilustración 4: Fases de prueba en el proceso de creación o evolución de un software



Fuente: Ian Sommerville (2005). *Ingeniería del Software*, página 75, Séptima Edición.

1.1.4 Ventajas y desventajas

Ventajas

- RUP es una metodología adaptable a cualquier proyecto.
- Existe un involucramiento de las partes interesadas como usuarios y desarrolladores continua.
- Fácil de dar seguimiento ya que tiene una visualización muy gráfica del proyecto y se realiza incrementalmente.
- Se basa en las mejores prácticas de ingeniería de software^[1].
- Garantiza la calidad del producto final.

Desventajas

- Para proyectos pequeños puede ser muy pesado y tedioso.
- Requiere conocimiento previo sobre UML.

^[1] Ingeniería de software: Es una disciplina que se ocupa del desarrollo, la operación y el mantenimiento de software.

- Puede generar trabajo adicional.
- Los estimados casi nunca se cumplen sobre todo los estimados en tiempo y recursos monetarios.

1.1.5 Aplicación de RUP en el desarrollo

La metodología de desarrollo RUP se enfoca en cuatro puntos esenciales que son conocidas como las cuatro P:

- **Personas:** Son todos los individuos involucradas dentro del desarrollo del proyecto como arquitecto, usuarios, desarrolladores, etc.
- **Proyecto:** Hace referencia a la organización involucrada en el desarrollo del producto.
- **Producto:** Son todos los entregables o artefactos que se entregan durante el tiempo de vida del proyecto (Ejemplo: Casos de uso, líneas de código, etc.).
- **Proceso:** Hace referencia al conjunto de actividades que se realizan para transformar necesidades en un producto.

Todo esto se realiza a través de buenas prácticas de ingeniería de software para garantizar la calidad del producto final, aquí hacemos referencia a seis prácticas fundamentales:

- Desarrollar software de forma iterativa.
- Utilizar arquitecturas basadas en componentes.
- Gestionar los requerimientos del cliente.
- Modelar el software visualmente.
- Verificar la calidad del software en todo el tiempo de vida de desarrollo.

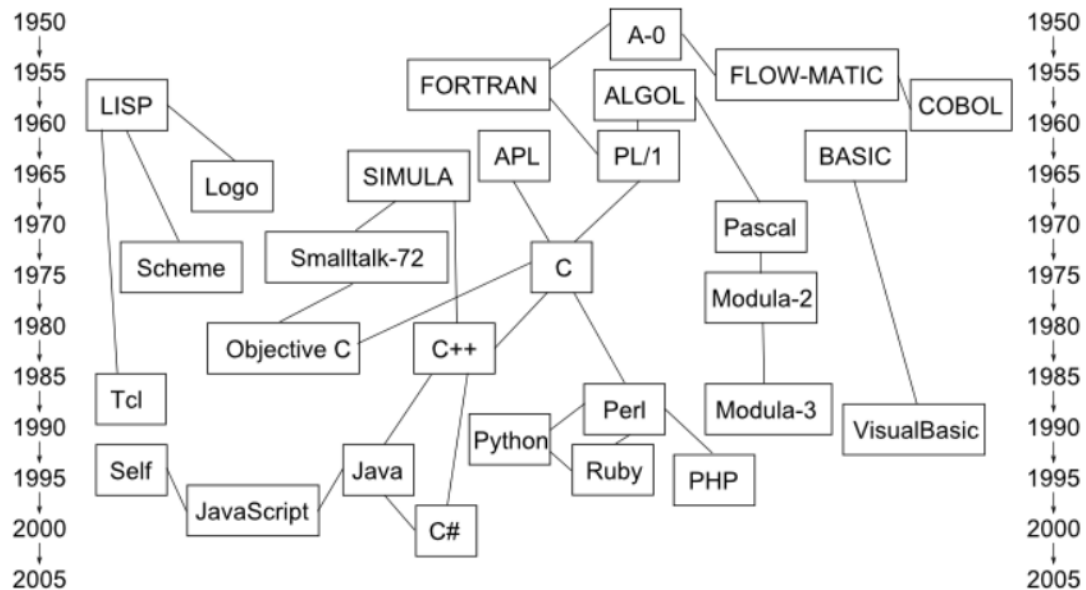
- Controlar los cambios que se le realice al software.

1.2 Lenguaje de programación Java

1.2.1 Introducción

Los lenguajes de programación^[1] fueron creados para facilitar el desarrollo de programas ejecutables^[2] los cuales son los encargados de asignar tareas que los ordenadores cumplen. Ahora es muy común encontrar ordenadores dentro de cualquier dispositivo es por esto que la evolución de los lenguajes de programación era necesaria como observamos en la siguiente imagen.

Ilustración 5: Evolución de los lenguajes de programación



Fuente: http://www.mclibre.org/consultar/python/otros/in_lenguajes_programacion.html

^[1] Lenguaje de programación: Son todos los lenguajes formales diseñados para realizar procesos que puedan ser interpretados por los ordenadores.

^[2] Programas ejecutables: Son programas los cuales pueden ser llevados a cabo por una computadora.

Cada lenguaje de programación tiene sus propias reglas semánticas, sintácticas, palabras reservadas y tipo de programación ya sea modular, secuencial u orientada a objetos. Como podemos observar en el grafico 5 la descendencia de un nuevo lenguaje siempre viene dado por otro o varios, como por ejemplo Java resulta de la descendencia de C++ el cual resulta de la descendencia de C.

1.2.2 Historia del lenguaje de programación Java

Java es un lenguaje creado en los 90s por James Gosling quien consiguió patrocinio de parte de Sun Microsystems^[k] para un proyecto llamado Green.

Sun Microsystems fue adquirida por Oracle^[l] Corporation en el 2010 formando parte de uno de los íconos de Silicon Valley.

Java fue publicado en 1995 como un componente fundamental en la plataforma Java. Java tiene derivación de los lenguajes C y C++, sus predecesores; es un lenguaje orientado a objetos compilado y multiplataforma, es decir que al tener su propia plataforma no importa el sistema operativo en que se lo corra.

El lenguaje es tan popular en el mercado que grandes empresas como Twitter, CERN, NYSE, London Multi Asset eXchange e incluso Ebay trabajan sobre él.

Java además de ser un software libre también da la posibilidad de elegir que proveedor consideremos, esto lo logró creando estándares sobre el lenguaje para quienes estuvieran interesados en comercializarlo así tenemos a los tres principales proveedores: Oracle, Red Hat e IBM.

^[k] Sun Microsystems: Era una compañía Norte Americana dedicada a la venta de computadores y componentes de computadores.

^[l] Oracle Corporation: Es una compañía de software que desarrolla base de datos.

Al ser un lenguaje que abarca tantas características sobre varios temas se decide subdividir la plataforma en tres las cuales son:

- Java SE: Hace referencia a todos los applets y aplicaciones de escritorio.
- Java EE: Hace referencia a aplicaciones WEB y Enterprise.
- Java ME: Trata todo lo que son dispositivos embarcados, Blu Ray players, dumbphones.

1.2.3 Características fundamentales

Java es un lenguaje orientado a objetos que tiene su propia máquina virtual que permite ejecutar código compilado Java el cual se basa en cuatro pilares fundamentales los cuales son:

- Abstracción.
- Encapsulamiento.
- Herencia.
- Polimorfismo.

Java también es un lenguaje de tipado estático, que es robusto esto es gracias a la evolución que tuvo del lenguaje C++ eliminando la existencia de punteros y trabajando con referencias, con una arquitectura neutral multihilo que contiene su propio recolector de basura (Garbage Collection) lo que nos brinda mayor optimización de memoria, además es un lenguaje portable al tener su propia plataforma sobre la cual corre.

1.2.4 Ventajas y desventajas

Ventajas

- Existen varios IDEs (entornos de desarrollo) o plataformas gratuitas para poder desarrollar código Java.
- La reutilización de código es una de las ventajas de este lenguaje.
- Este lenguaje tiene una gran demanda en el mercado.
- En el internet se puede encontrar bastantes fuentes de código gratuito.
- Este lenguaje nos proporciona una fácil conexión con diferentes motores de bases de datos.
- Existen amplias librerías de código para todo tipo de programas.
- POO o programación orientada a objetos.

Desventajas

- Es un lenguaje compilado e interpretado lo cual necesita de un intérprete y esto hace que el proceso de traducción que realiza el intérprete tome más tiempo.
- Existen plugins o herramientas que no son gratuitas.
- Los compiladores e intérpretes generan más código máquina y esto hace que el código sea de mayor tamaño.

1.2.5 Aplicaciones de Java

En este lenguaje existen diferentes tipos de aplicaciones como las siguientes:

Aplicación: “Una aplicación se ejecuta como una aplicación en cualquier plataforma operativa además puede tener una interfaz gráfica específica e interrelacionarse con otros archivos” (Maury Camilo, 2010, pp: 1)

Esta es más amplia que las demás, se compila con el archivo Javac y luego se genera un archivo .class el cual es interpretado.

Applet: “Un applet no corre en cualquier entorno operativo, más bien solo se ejecuta en un navegador pero también puede contar con una interfaz gráfica.” (Maury Camilo, 2010, pp: 1)

Estos están limitados al navegador y se usan cuando existe una función en una página web.

Servlets: “Son aplicaciones que se ejecutan en un servidor de aplicaciones web y que como resultado de su ejecución resulta una página web.” (Jorge Sánchez, 2014, pp: 5)

1.3 Base de datos PostgreSQL

1.3.1 Introducción

PostgreSQL es un sistema de bases de datos relacionales orientadas a objetos que fue desarrollada en California en el departamento de ciencias computacionales de la Universidad de Berkeley aunque en ese entonces su nombre era Postgres y en el año 1994 abandonó ese nombre y tomó el del actual gestor de bases de datos PostgreSQL .

PostgreSQL fue el primero en contener muchos conceptos que no tenían otros sistemas de bases de datos comerciales hasta mucho tiempo después.

PostgreSQL tiene licencia libre y esto permite que cualquier usuario lo pueda usar, modificar y distribuir de manera libre y gratuita como para propósitos comerciales, privados o académicos.

1.3.2 Breve historia de PostgreSQL

Este sistema de base de datos se desarrolló con el nombre Postgres en 1986 como un proyecto en la Universidad de Berkeley en California. Postgres fue creado con la orientación a objetos y distintas funciones avanzadas de bases de datos.

Postgres tenía su propio lenguaje de consultas que se llamaba Postquel, siendo superior al dominante SQL pero existía un problema y era que este no se alineaba con las necesidades de la industria, que ya se había estandarizado con SQL. Por esa razón, en 1995 se reemplazó Postquel con un subconjunto extendido de SQL.

En 1996 se cambió el nombre de Postgres a PostgreSQL, el cual se mantiene hasta la actualidad. En ese mismo año se lanzó la versión 6.0. Desde ese entonces se liberan las versiones consecutivamente partiendo de la versión 6.0

1.3.3 Características fundamentales

PostgreSQL posee diferentes características que lo hacen un sistema de bases de datos fuerte y amplio en el mercado.

Características:

- Tiene licencia libre, no tiene costo, por lo que la instalación es ilimitada

- PostgreSQL ha demostrado ser confiable y estable
- Es de código abierto por lo que el usuario puede adaptarlo a su necesidad
- Soporta varias plataformas es decir es un sistema multiplataforma
- Soporta varios lenguajes de programación C, C++, Delphi, Python, Java, PHP, entre otros.
- Cuenta con herramientas gráficas de diseño y herramientas de administración de base de datos como el pgAdmin.
- Soporta conectividad TCP/IP y también conexiones ODBC y JDBC
- Maneja varios tipos de datos
- Es orientado a objetos
- Soporta triggers, procedimientos almacenados, vistas, claves foráneas, consultas y subconsultas

1.3.4 Ventajas y desventajas

Ventajas

- Es multiplataforma, al ser desarrollado en el lenguaje C, este puede ser compilado en la mayoría de los sistemas operativos existentes.
- Su licencia es libre.
- Además de manejar los tipos de datos habituales también puede manejar objetos binarios grandes como imágenes, sonido y video.
- Permite el uso de caracteres internacionales y utiliza Unicode.
- Permite realizar copias de seguridad en línea.
- Es escalable en cuanto a cantidad de datos a administrar y a número de usuarios concurrentes a manejar.

Desventajas

- Consume más recursos que MySQL por lo que se necesitan mayores características de hardware para ejecutarlo.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.
- Instalación Ilimitada
- Es vulnerable sin protección adecuada

1.3.5 Aplicaciones de PostgreSQL

PostgreSQL puede ser utilizado para el desarrollo de diferentes aplicaciones que realicen diferentes procesos como los siguientes:

- Realizar sentencias SQL como consultar datos, insertar datos, borrar datos, actualizar datos, etc.
- Crear vistas (conjunto de registros que son resultado de una consulta realizada, son similares a las tablas)
- Realizar transacciones (conjunto de acciones que se ejecutan todas ellas o ninguna). Con las transacciones se asegura que los datos sean consistentes
- Realizar procedimientos almacenados (código ejecutable que evita las transferencias innecesarias optimizando de esta manera las aplicaciones)

CAPÍTULO 2

2. FASE DE INICIO

2.1 Levantamiento de los principales procesos

2.1.1 Cadena de valor

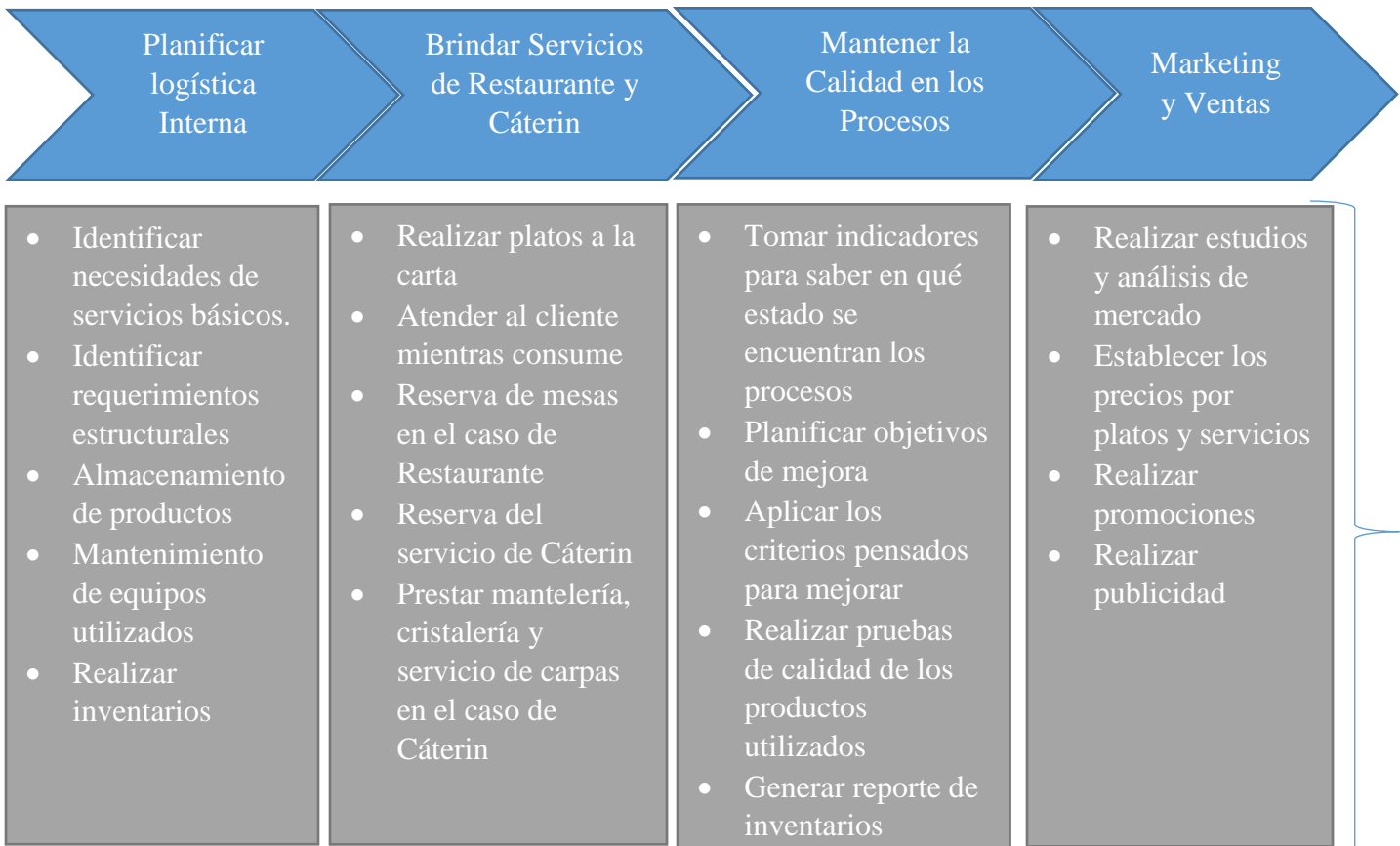
Ilustración 6: Cadena de valor del Restaurante y Servicios de Cáterin Alexander



Fuente: Jorge Cunalata & David Morán

2.1.2 Identificación principales procesos

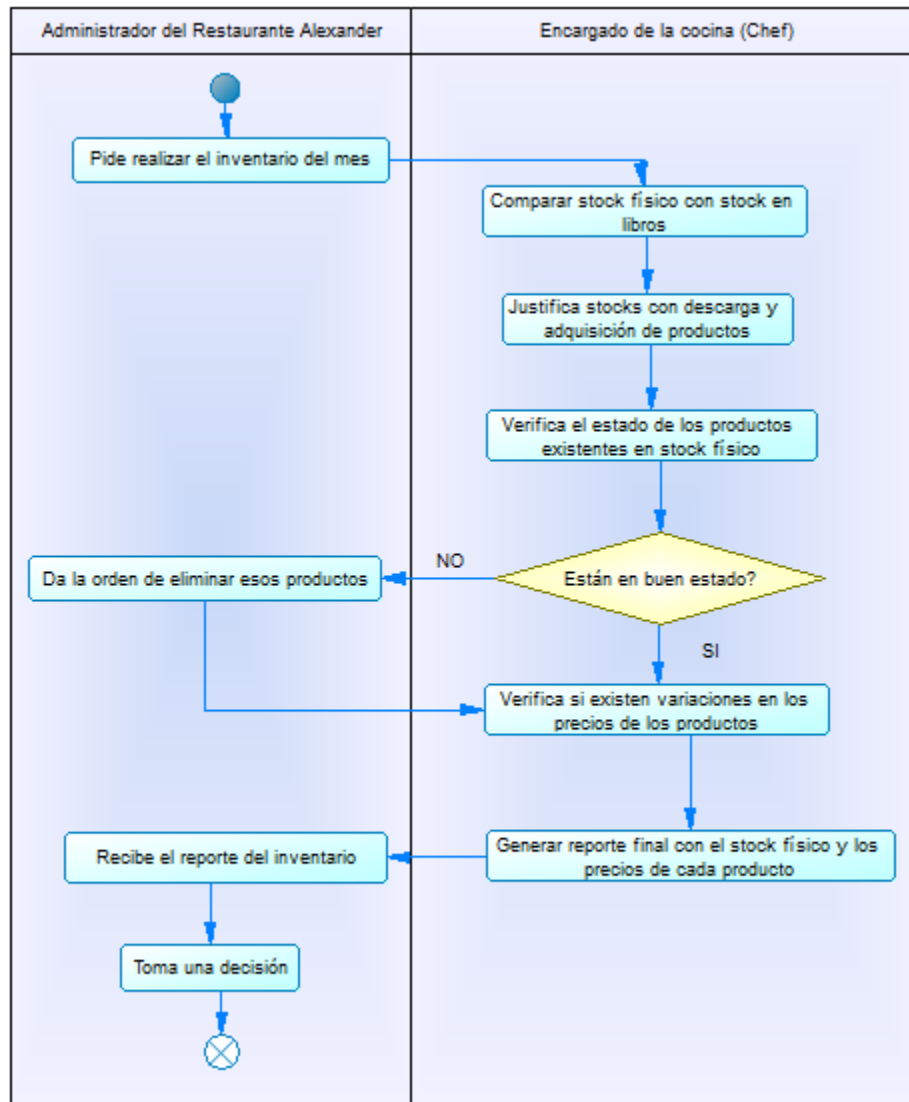
Ilustración 7: Procesos principales y subprocesos del Restaurante y Servicios de Cáterin Alexander



Fuente: Jorge Cunalata & David Morán

2.1.3 Diagramación y descripción del proceso de inventarios

Ilustración 8: Diagrama del proceso de Inventario



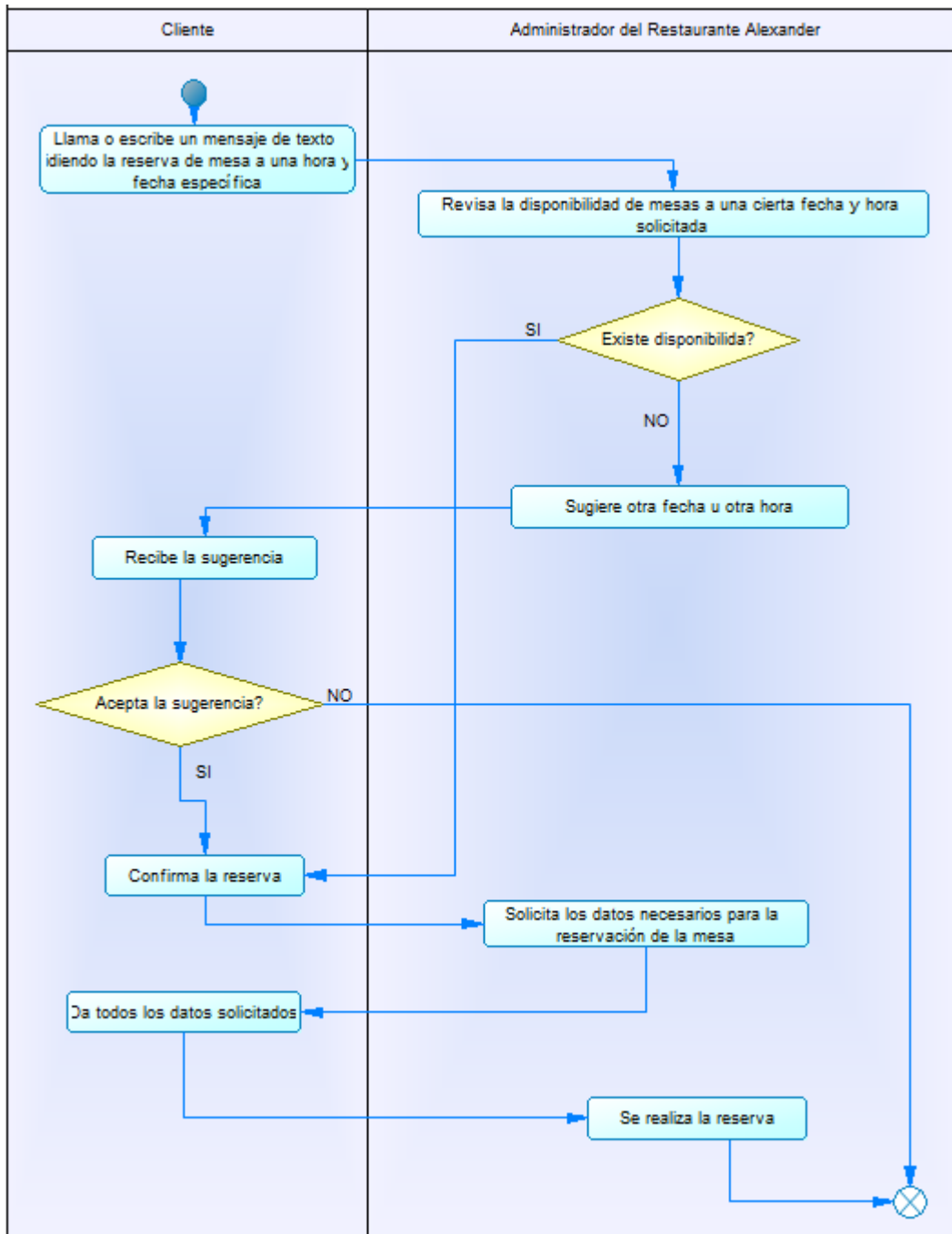
Fuente: Jorge Cunalata & David Morán

1. El administrador del Restaurante Alexander solicita se realice el reporte de inventario al final de cada mes para tomar decisiones sobre su stock de productos.
2. El encargado de la cocina que en el caso del Restaurante Alexander es el Chef realiza una comparación del stock físico y el stock en libros que se posee al final del mes.
3. El encargado de la cocina (Chef) justifica el stock en físico con la descarga del producto que tuvieron contra la adquisición del mismo en dicho mes.

4. Se verifica el estado de los productos existentes en el stock físico, estos podrían estar en buen estado (apto para el consumo humano o almacenar) o mal estado (no apto para el consumo humano).
5. En el caso que existan productos en mal estado dentro del inventario físico el administrador del Restaurante Alexander da la orden de eliminar estos productos.
6. Ya una vez eliminados los productos en mal estado del inventario físico el encargado de la cocina (Chef) verifica si existen variaciones en los precios de los productos.
7. El encargado de la cocina (Chef) genera el reporte final con el stock físico real y los precios de cada producto.
8. El administrador del Restaurante Alexander recibe el reporte del inventario.
9. El administrador del Restaurante Alexander toma una decisión.

2.1.4 Diagramación y descripción del proceso de gestión de reserva de mesas

Ilustración 9: Diagrama del proceso de Inventario



Fuente: Jorge Cunalata & David Morán

1. El cliente se comunica con el administrador del Restaurante Alexander mediante un mensaje de texto o una llamada pidiendo la reserva de una mesa a una fecha y hora específica.
2. El administrador del Restaurante Alexander revisa la disponibilidad de mesas a una cierta fecha y hora solicitada por el cliente.
3. **EXISTEN DOS FLUJOS ALTERNOS SOBRE LA DISPONIBILIDAD DE MESAS A LA FECHA Y HORA SOLICITADA POR EL CLIENTE, ESTOS SON**
 - 3.A. Si no existe la disponibilidad de una mesa a esa hora y fecha solicitada por el cliente el administrador del Restaurante Alexander sugiere otra fecha u otra hora para realizar la reserva.

EXISTEN DOS FLUJOS ALTERNOS SOBRE LA ACEPTACIÓN DE LA SUGERENCIA DEL ADMINISTRADOR, ESTOS SON:

- 3.A.1. El cliente no acepta la sugerencia del administrador del restaurante y se termina el proceso de reserva de mesas.
- 3.A.2 El cliente si acepta la sugerencia del administrador del restaurante y confirma la reserva de mesa con la hora y fecha sugerida.

FIN DEL FLUJO ALTERNO

- 3.B. El cliente confirma la reserva de la mesa al existir la disponibilidad de la misma a la hora y fecha solicitada.

FIN DEL FLUJO ALTERNO

4. El administrador del Restaurante Alexander solicita los datos del cliente para crear la reservación de mesa.
5. El cliente da todos los datos solicitados por el administrador.
6. El administrador realiza la reservación de la mesa.

2.2 Modelado del negocio

2.2.1 Análisis de situación actual del negocio

Introducción

El Restaurante y Servicios de Cáterin Alexander es la organización encargada de brindar un servicio de restaurante al público en general y también está encargado de brindar un servicio de cáterin a empresas situadas en el sector del aeropuerto Internacional Mariscal Sucre.

Propósito

El Restaurante y Servicios de Cáterin Alexander busca introducir el servicio de reserva de mesas en el caso de restaurante para una mejor experiencia de los clientes y automatizar el proceso de inventario para tener un mejor control sobre los productos y los proveedores.

Objetivos del negocio

- Monopolizar el mercado de restaurante y cáterin del sector del aeropuerto Internacional Mariscal Sucre.

- Ofrecer platos de alta calidad con productos frescos para la máxima satisfacción del cliente.
- Contar con un espacio adecuado para la atención del cliente.
- Contar con una excelente atención al cliente.

Beneficios del negocio:

- Entregar platos saludables y de buena calidad.
- Brindar un servicio personalizado para crear una experiencia positiva para el cliente.
- Brindar platos con costos accesibles al cliente promedio.

2.2.2 Descripción breve del sistema

El sistema para reserva de mesas y automatización del proceso de inventario que se desarrollará tiene dos objetivos principales. El primero es realizar reserva de mesas para brindar una mejor experiencia al cliente evitando que este tenga que esperar por una mesa en el caso que se encuentre lleno el local.

Como segundo objetivo principal tenemos la automatización del proceso de inventario y generación de reportes del mismo, esto ayudará al administrador del local a saber el stock y estado de los productos para poder tomar decisiones y prevenir futuros inconvenientes al momento de realizar los platos para la venta. Los reportes generados podrán ser registrados por el administrador como datos históricos a ser analizados y así tener una mejor visión del negocio en cuanto a su inventario de producto y proveedores.

2.2.3 Descripción breve de las necesidades del negocio en el sistema propuesto

Previo a la propuesta del desarrollo de este sistema el administrador del Restaurante Alexander nos manifestó la necesidad que tiene de llevar el inventario de una manera más

correcta y segura también nos explicó que sería de gran utilidad llevar un control sobre el estado de los productos de acuerdo con su fecha límite de caducidad ya que existe mucho desperdicio de alimentos por no ser consumidos a tiempo.

Nos comentaron que quieren implementar nuevos servicios como reserva de mesas para aumentar la satisfacción de los clientes, esta necesidad surge ya que ha habido varios casos en el que el restaurante se encuentra lleno en determinadas horas y han perdido clientes por esta situación, lo que se pretende mediante la automatización de reserva de mesas es distribuir el número de clientes de una mejor manera impidiendo así que se tenga pérdida de clientes al no tener mesas disponibles.

2.2.4 Beneficios y mejoras que producirá el sistema

Ya identificados los principales problemas que presenta el Restaurante Alexander lo que buscamos con el sistema a desarrollar es dar un valor agregado al servicio que ofrecen actualmente.

El sistema será capaz de llevar un control sobre el inventario más realista y seguro optimizando los recursos dentro del stock físico que posee el Restaurante Alexander. También será capaz de llevar un control sobre el estado de los productos adquiridos evitando el desperdicio de los mismos al ser consumidos antes de su fecha de caducidad.

El sistema será capaz de llevar un control sobre la reserva de mesas que mejorará la experiencia de los clientes al no tener que esperar si el local se encuentra lleno, también permite que el administrados pueda distribuir de una mejor manera el número de clientes en determinadas horas "pico" evitando así pérdida del mayor número de clientes.

2.3 Requerimientos

2.3.1 Descripción general del sistema

El sistema de gestión de del restaurante "Alexander" permite registrar las reservaciones de mesas al administrador del restaurante y también permite la automatización del proceso de inventarios. El usuario administrador tiene todos los permisos mientras que el usuario cliente solo puede seleccionar la mesa y el horario de reserva de mesa.

El usuario administrador es el único que puede ingresar la ficha de reserva de mesa para el usuario cliente con los datos de reserva.

El usuario administrador es el encargado de ingresar los productos y la información necesaria de dichos productos los cuales son: fecha de ingreso, fecha de caducidad, stock, proveedores y precios.

El sistema se encargará de presentar visualmente al administrador la ubicación y el número de mesas disponibles como el de las mesas ocupadas, también este sistema se encargará de anular la reserva de una mesa si el usuario cliente tiene un retraso de 15 minutos de llegada permitiendo así el uso de esa mesa a otro cliente.

El sistema se encargará de informar al usuario administrador con un panel de alerta cuando cualquier producto dentro del inventario esté a punto de vencer su fecha de caducidad o cuando este esté quedándose sin stock.

El sistema realizará estos procesos mediante una interfaz gráfica la cual será amigable con el usuario y será instalada en las computadoras del restaurante.

2.3.2 Requerimientos funcionales

Dentro de los requerimientos funcionales del sistema se encuentran los siguientes:

- **Administración de mesas:** En este requerimiento el usuario administrador está a cargo de ingresar el registro de reserva de una mesa con la información que le proporcione el usuario cliente.
- **Administración del inventario:** En este requerimiento el usuario administrador está a cargo de ingresar los productos y su respectiva información al momento de que estos sean adquiridos.
- **Reporte de inventario:** Esta funcionalidad le permite al usuario administrador generar un reporte mensual del stock del inventario y el estado del producto.

2.3.3 Requerimientos de hardware y software

Requerimientos de software:

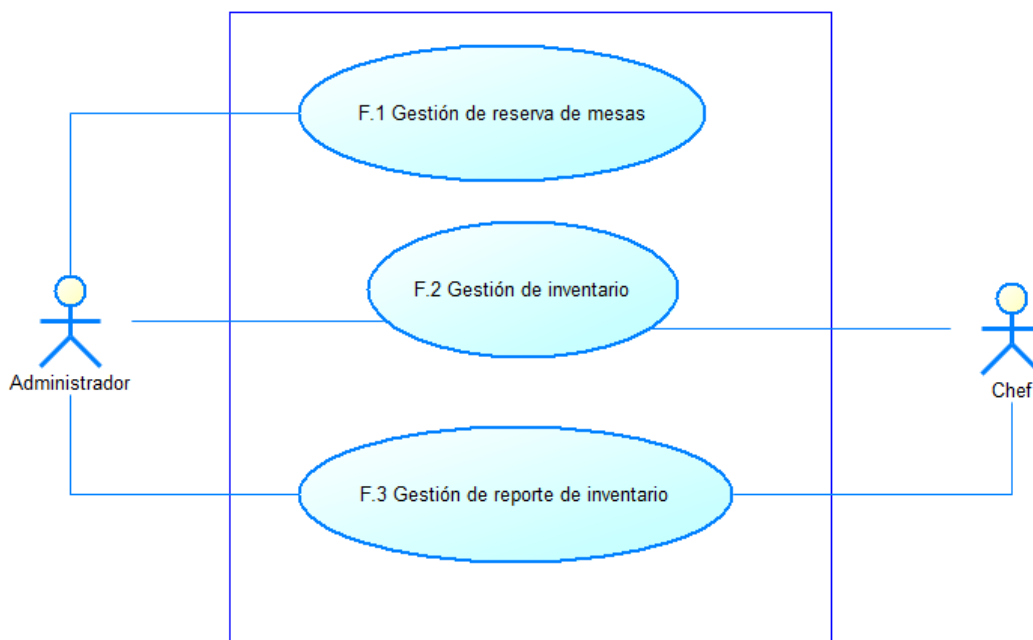
- Windows Vista o superior
- Java 8 o superior
- PostgreSQL 9.4 o superior
- Requerimientos de Hardware
- 2 GB de RAM o superior
- Puerto usb 2.0 o superior
- Unidad de CD/DVD-ROM

- Disco duro 512 MB de memoria disponible
- Una impresora para imprimir los reportes

2.3.4 Diagrama de casos de uso

Diagrama general de caso de uso de los procesos a automatizar

Ilustración 10: Diagrama de Casos de Uso General

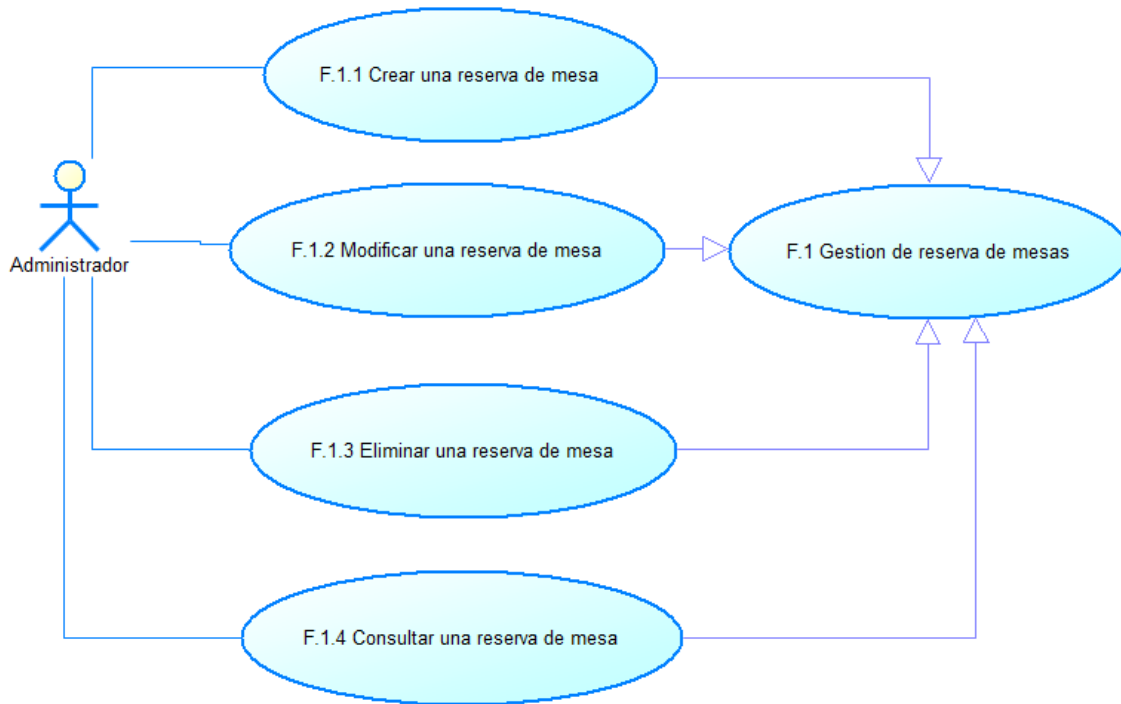


Fuente: Jorge Cunalata & David Morán

F1. Gestión de Reserva de Mesas

Esta funcionalidad permite crear la reserva de mesas, cada reserva tiene su propio código de reserva único y contiene los datos proporcionados por el cliente, además nos permite modificar y eliminar la reserva.

Ilustración 11: Diagrama de Casos de Uso Gestión de Reserva de Mesas



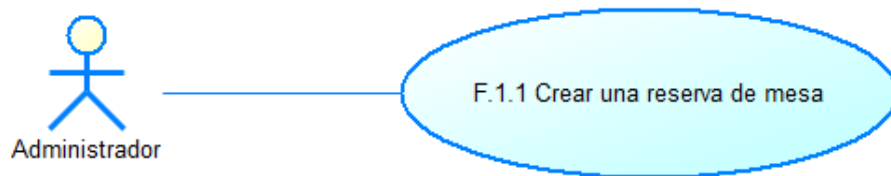
Fuente: Jorge Cunalata & David Morán

F1.1 Crear una reserva de mesa

Esta funcionalidad le permite al administrador del Restaurante Alexander crear una reserva de mesa.

Actores: Administrador

Ilustración 12: Diagrama de Casos de Uso Crear Una Reserva de Mesa



Flujo Principal:

- 1.- El actor presiona crear una reserva de mesa.
- 2.- El sistema verifica la disponibilidad de mesas.
- 3.- El sistema despliega una ventana con los campos requeridos para crear una reserva de mesa.
- 4.- El actor ingresa los datos.
- 5.- El sistema valida los datos ingresados en los campos requeridos.
- 6.- El actor presiona el botón CREAR.
- 7.- El sistema almacena los datos ingresados en la base de datos. (E1)
- 8.- El sistema presenta una ventana de alerta “Reserva Creada Exitosamente”.

Flujo Alternativo:

- 2.- Si no existe disponibilidad de mesa el sistema muestra un mensaje de alerta “No hay mesas disponibles”.

Excepciones:

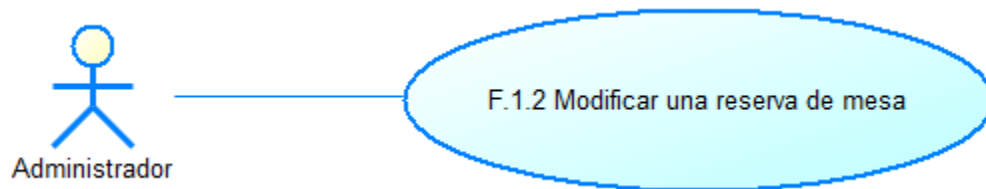
- 1.- Problemas con la conexión de la base de datos.
- Solución: Intentar más tarde o contactar al técnico.

F1.2 Modificar una reserva de mesa

Esta funcionalidad le permite al administrador del Restaurante Alexander modificar los datos de una reserva de mesa.

Actores: Administrador

Ilustración 13: Diagrama de Casos de Uso Modificar Una Reserva de Mesa



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona el botón Modificar Reserva.
- 2.- El sistema presenta las reservas existentes.
- 3.- El actor selecciona la reserva a modificar.
- 4.- El sistema presenta los campos con los datos registrados previamente.
- 5.- El actor cambia los valores de los campos que se quiere modificar.
- 6.- El sistema valida los datos ingresados en los campos modificados.
- 7.- El actor presiona el botón Guardar.

- 8.- El sistema almacena los datos ingresados en la base de datos. (E1)
- 9.- El sistema presenta una ventana de alerta “Reserva Modificada Exitosamente”.

Excepciones:

- 1.- Problemas con la conexión de la base de datos.

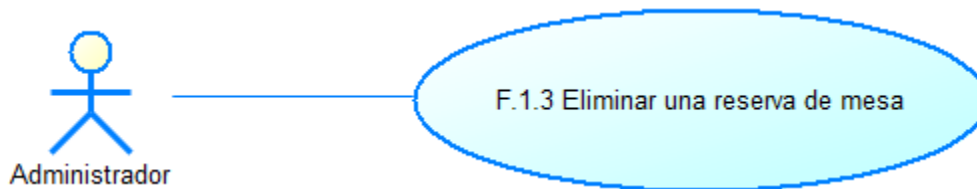
Solución: Intentar más tarde o contactar al técnico.

F1.3 Eliminar una reserva de mesa

Esta funcionalidad le permite al administrador del Restaurante Alexander eliminar una reserva de mesa.

Actores: Administrador

Ilustración 14: Diagrama de Casos de Uso Eliminar Una Reserva de Mesa



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona el botón Eliminar Reserva.
- 2.- El sistema presenta las reservas existentes.
- 3.- El actor selecciona la reserva a eliminar.
- 4.- El sistema presenta los campos con los datos registrados previamente.

- 5.- El actor presiona el botón Eliminar.
- 6.- El sistema muestra un mensaje de alerta “Está Seguro”.
- 7.- El actor presiona el botón Si.
- 8.- El sistema elimina el registro de la base de datos. (E1)
- 9.- El sistema muestra un mensaje de alerta “Reserva Eliminada Exitosamente”.

Flujo Alternativo:

- 7.- El actor presiona el botón Cancelar.

Excepciones:

- 1.- Problemas con la conexión de la base de datos.

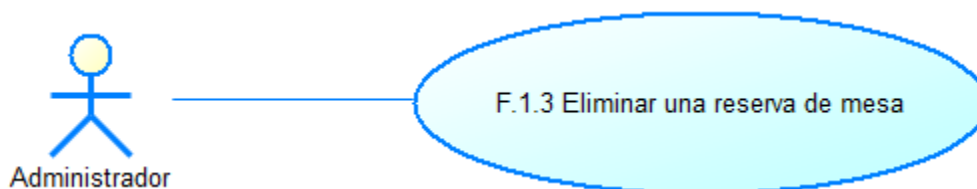
Solución: Intentar más tarde o contactar al técnico.

F1.4 Consultar una reserva de mesa

Esta funcionalidad le permite al administrador del Restaurante Alexander consultar una reserva de mesa.

Actores: Administrador

Ilustración 15 Gráfico 14: Diagrama de Casos de Uso Consultar Una Reserva de Mesa



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona el botón Ventana de Reserva de Mesas.
- 2.- El sistema presenta la ventana de Reserva de Mesas.
- 3.- El actor ingresa la fecha de la reserva a consultar.
- 4.- El sistema verifica si existe la reserva en la fecha ingresada.
- 5.- El sistema presenta el Id de la Reserva, la fecha, la hora, la cantidad de personas de la reserva.

Flujo Alterno:

- 5.- El sistema presenta un mensaje de que en aquella fecha no existen reservas.

Excepciones:

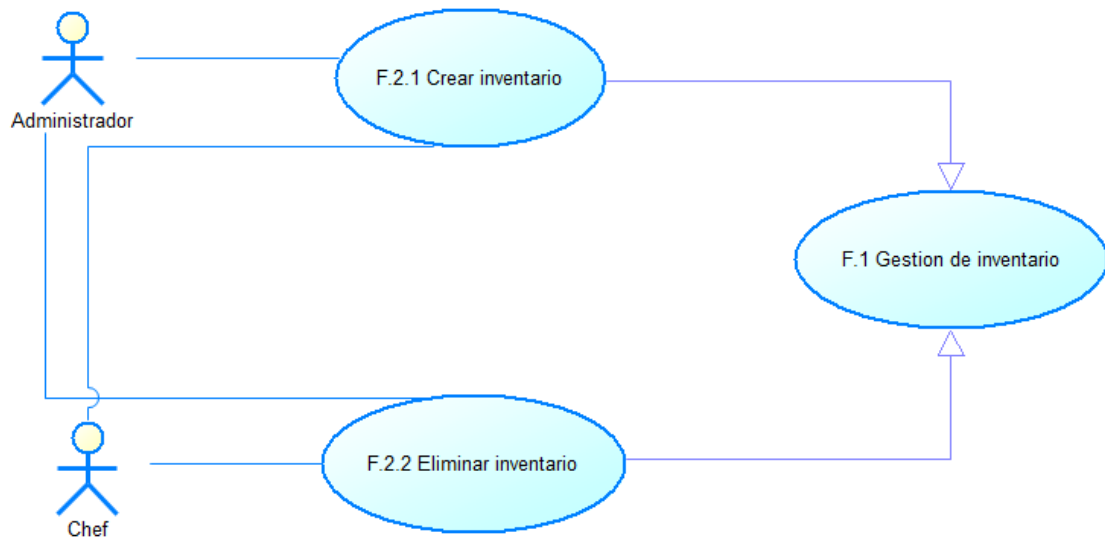
- 1.- Problemas con la conexión de la base de datos.

Solución: Intentar más tarde o contactar al técnico.

F2. Gestión de Inventario

Esta funcionalidad permite crear un inventario en un tiempo determinado o solicitado por el administrador, cada inventario contiene stock físico de productos y sus respectivos precios, además esta funcionalidad nos permite modificar y eliminar los inventarios.

Ilustración 16: Diagrama de Casos de Uso Gestión de Inventario



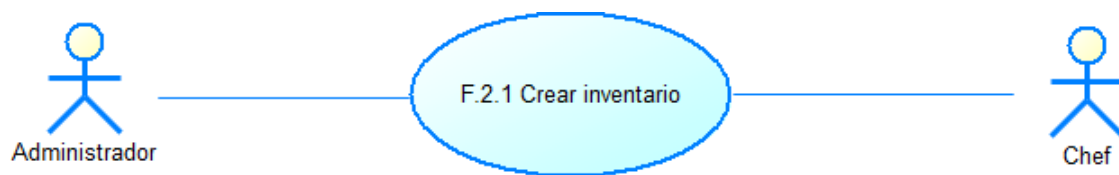
Fuente: Jorge Cunalata & David Morán

F2.1 Crear Inventario

Esta funcionalidad le permite al administrador del Restaurante Alexander crear un inventario

Actores: Administrador, Chef

Ilustración 17: Diagrama de Casos de Uso Crear Inventario



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona crear inventario.

- 2.- El sistema despliega una ventana con los campos requeridos para crear un inventario.
- 3.- El actor ingresa los datos.
- 4.- El sistema valida los datos ingresados en los campos requeridos.
- 5.- El actor presiona el botón CREAR.
- 6.- El sistema almacena los datos ingresados en la base de datos. (E1)
- 7.- El sistema presenta una ventana de alerta “Inventario Creado Exitosamente”.

Excepciones:

- 1.- Problemas con la conexión de la base de datos.

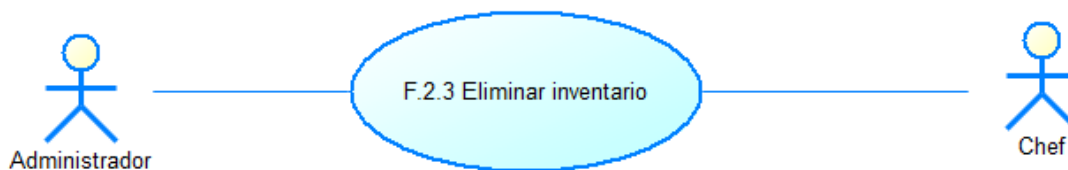
Solución: Intentar más tarde o contactar al técnico.

F2.2 Eliminar inventario

Esta funcionalidad le permite al administrador del Restaurante Alexander eliminar un inventario.

Actores: Administrador, Chef.

Ilustración 18: Diagrama de Casos de Uso Eliminar Inventario



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona el botón Eliminar Inventario.
- 2.- El sistema presenta los inventarios existentes.

- 3.- El actor selecciona el inventario a eliminar.
- 4.- El sistema presenta los campos con los datos registrados previamente.
- 5.- El actor presiona el botón Eliminar.
- 6.- El sistema muestra un mensaje de alerta “Está Seguro”.
- 7.- El actor presiona el botón “Si”.
- 8.- El sistema elimina el registro de la base de datos. (E1)
- 9.- El sistema muestra un mensaje de alerta “Inventario Eliminado Exitosamente”.

Flujo Alternativo:

- 7.- El actor presiona el botón Cancelar.

Excepciones:

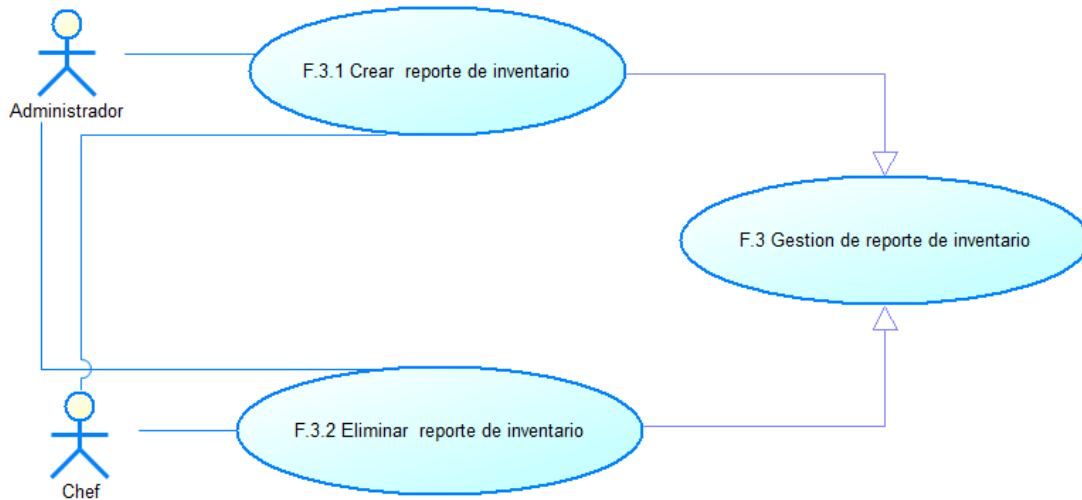
- 1.- Problemas con la conexión de la base de datos.

Solución: Intentar más tarde o contactar al técnico.

F3. Gestión de reporte de inventario

Esta funcionalidad permite crear y eliminar reportes de inventarios creados por el administrador o el chef.

Ilustración 19: Diagrama de Casos de Uso Gestión de Inventario



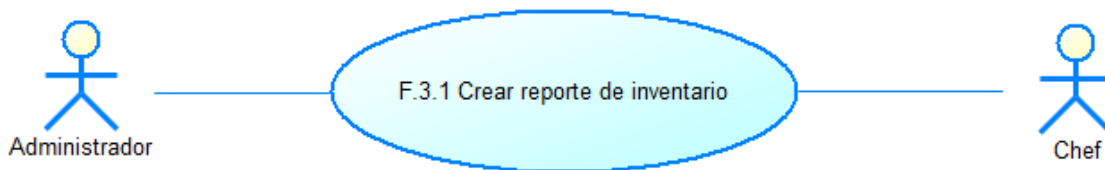
Fuente: Jorge Cunalata & David Morán

F3.1 Crear reporte de inventario

Esta funcionalidad le permite al administrador o chef del Restaurante Alexander crear un reporte de inventario.

Actores: Administrador, Chef

Ilustración 20: Diagrama de Casos de Uso Crear Inventario



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona crear reporte de inventario.
- 2.- El sistema despliega los inventarios existentes.
- 3.- El actor selecciona el inventario del cual requiere el reporte.
- 4.- El actor presiona el botón GENERAR REPORTE.
- 5.- El sistema genera el reporte del inventario.
- 6.- El sistema almacena el reporte en la base de datos. (E1)

Excepciones:

- 1.- Problemas con la conexión de la base de datos.

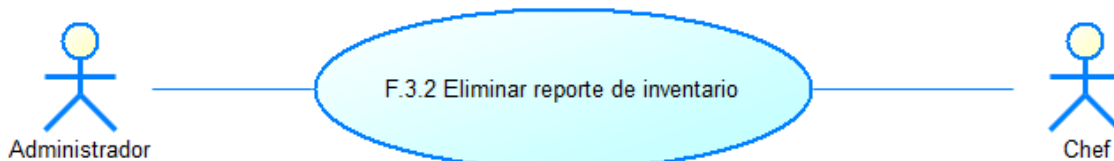
Solución: Intentar más tarde o contactar al técnico.

F1.2 Eliminar reporte de inventario

Esta funcionalidad le permite al administrador o chef del Restaurante Alexander eliminar un inventario ya existente.

Actores: Administrador, Chef.

Ilustración 21: Diagrama de Casos de Uso Eliminar Inventario



Fuente: Jorge Cunalata & David Morán

Flujo Principal:

- 1.- El actor presiona el botón Eliminar Reporte de Inventario.
- 2.- El sistema presenta los reportes existentes.
- 3.- El actor selecciona el reporte a eliminar.
- 4.- El actor presiona el botón Eliminar.
- 5.- El sistema muestra un mensaje de alerta “Está Seguro”.
- 6.- El actor presiona el botón “Si”.
- 7.- El sistema elimina el reporte de la base de datos. (E1)
- 8.- El sistema muestra un mensaje de alerta “Reporte Eliminado Exitosamente”.

Flujo Alternativo:

- 7.- El actor presiona el botón Cancelar.

Excepciones:

- 1.- Problemas con la conexión de la base de datos.

Solución: Intentar más tarde o contactar al técnico

CAPÍTULO 3.

3. FASE DE ELABORACIÓN

3.1 Análisis y diseño

3.1.1 Análisis del sistema actual

Al realizar el análisis de la situación actual del Restaurante y Servicio de Cáterin Alexander se encontró que existen algunas debilidades que se detallan a continuación:

- No cuentan con un proceso automatizado para la gestión de reserva de mesas. Este proceso es inexistente.
- No cuentan con un proceso automatizado para llevar el control de inventario. Este proceso lo realizan manualmente con hojas de control de inventario.
- No cuentan con un control de fechas de caducidad sobre productos perecibles.
- No cuentan con un control de stock de productos que alerte si la cantidad física existente en bodega satisface la demanda requerida.
- No cuentan con un proceso automatizado de generación de reporte de inventario.

Este proyecto está enfocado a un mejor control sobre los productos físicos dentro del stock en bodega tanto a su estado como a la cantidad óptima para satisfacer la demanda estimada.

El proyecto intenta implementar el proceso de reserva de mesas como una nueva funcionalidad del Restaurante y Servicio de Cáterin Alexander para brindar una mejor experiencia a los clientes y aumentar su competitividad dentro del mercado.

3.1.2 Estructura del producto

3.1.2.1 Diseño arquitectónico

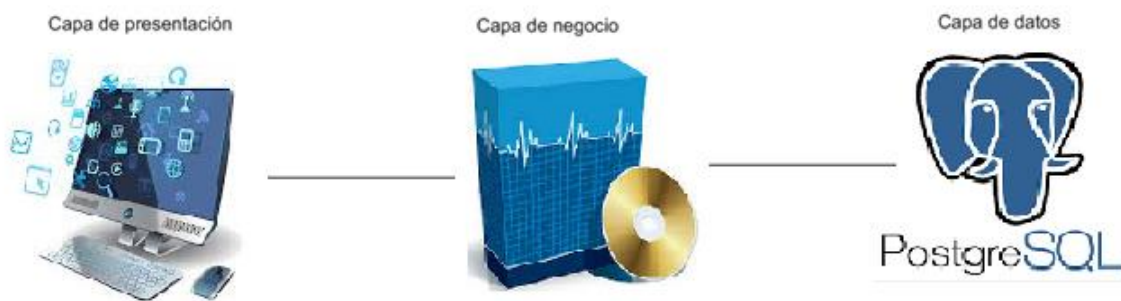
El sistema que contiene la automatización del proceso de inventario y el proceso de gestión de reserva de mesas para el Restaurante y Servicio de Cáterin Alexander será desarrollado como una aplicación de escritorio que permita al usuario (Administrador o Chef) realizar la gestión del proceso de reserva de mesas, gestión de inventario y gestión de reporte de inventario.

El aplicativo será implementado en la PC del administrador del restaurante.

La aplicación será completamente codificada en el lenguaje de programación Java, utilizando el IDE Netbeans 8.1 y se va a implementar en tres capas: Capa del Cliente, Capa Intermedia (lógica del negocio) y Capa de Datos.

- La capa de datos será implementada utilizando el gestor de base de datos PostgreSQL 9.3.4
- La capa intermedia será implementada utilizando Netbeans 8.1
- La Capa del Cliente se comunicará con la capa intermedia a través de la aplicación de escritorio.

Ilustración 22: Programacion en Capas



Fuente: Jorge Cunalata & David Morán

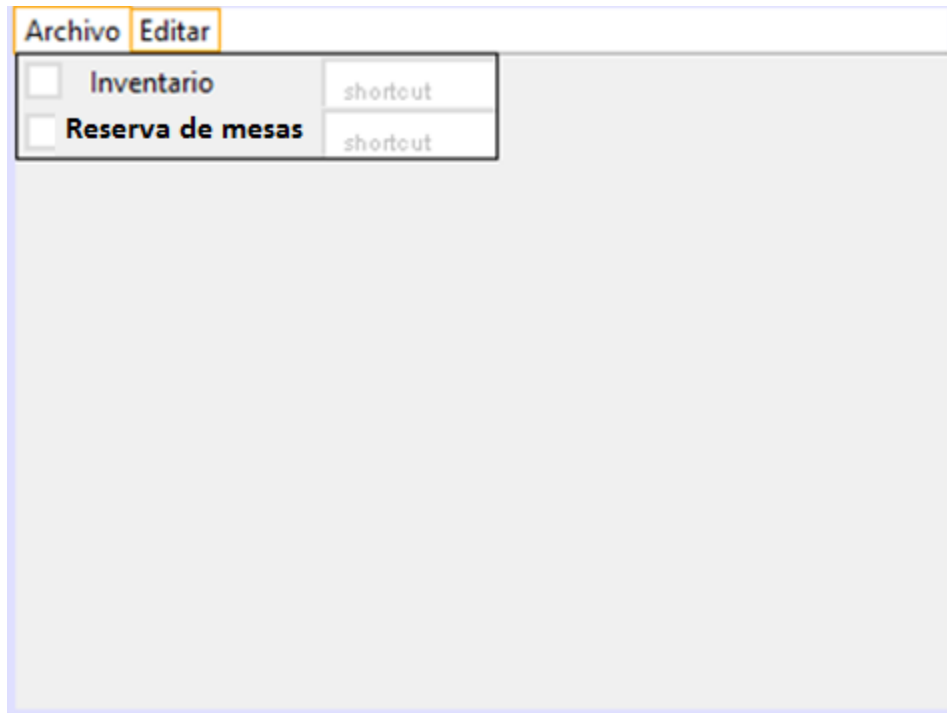
3.1.3 Diseño de interfaces

A continuación, se mostrarán el diseño de las interfaces a crear para el desarrollo de la aplicación.

Menú Principal

Es la pantalla principal donde se encontrarán todas las opciones para manejar el sistema y navegar en sus diferentes funciones.

Ilustración 23: Interfaz Menú Principal

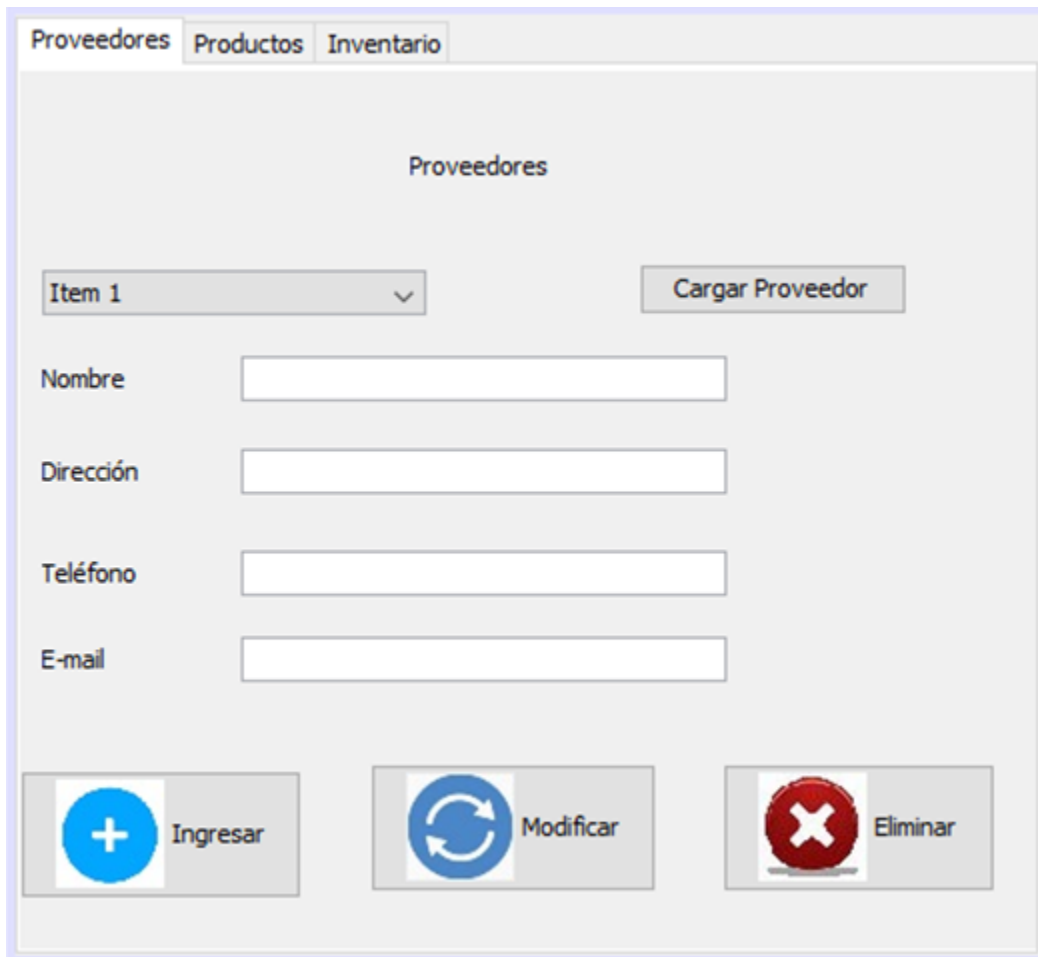


Fuente: Jorge Cunalata & David Morán

Pantalla Proveedores

Esta pantalla es para la creación, modificación, eliminación y consulta de los proveedores, se autogenera el id del proveedor, se ingresa el nombre, dirección, teléfono y su email.

Ilustración 24: Interfaz Proveedores



The screenshot displays a web application interface for managing providers. At the top, there are three tabs: 'Proveedores' (selected), 'Productos', and 'Inventario'. Below the tabs, the title 'Proveedores' is centered. The main area contains a form with the following elements:

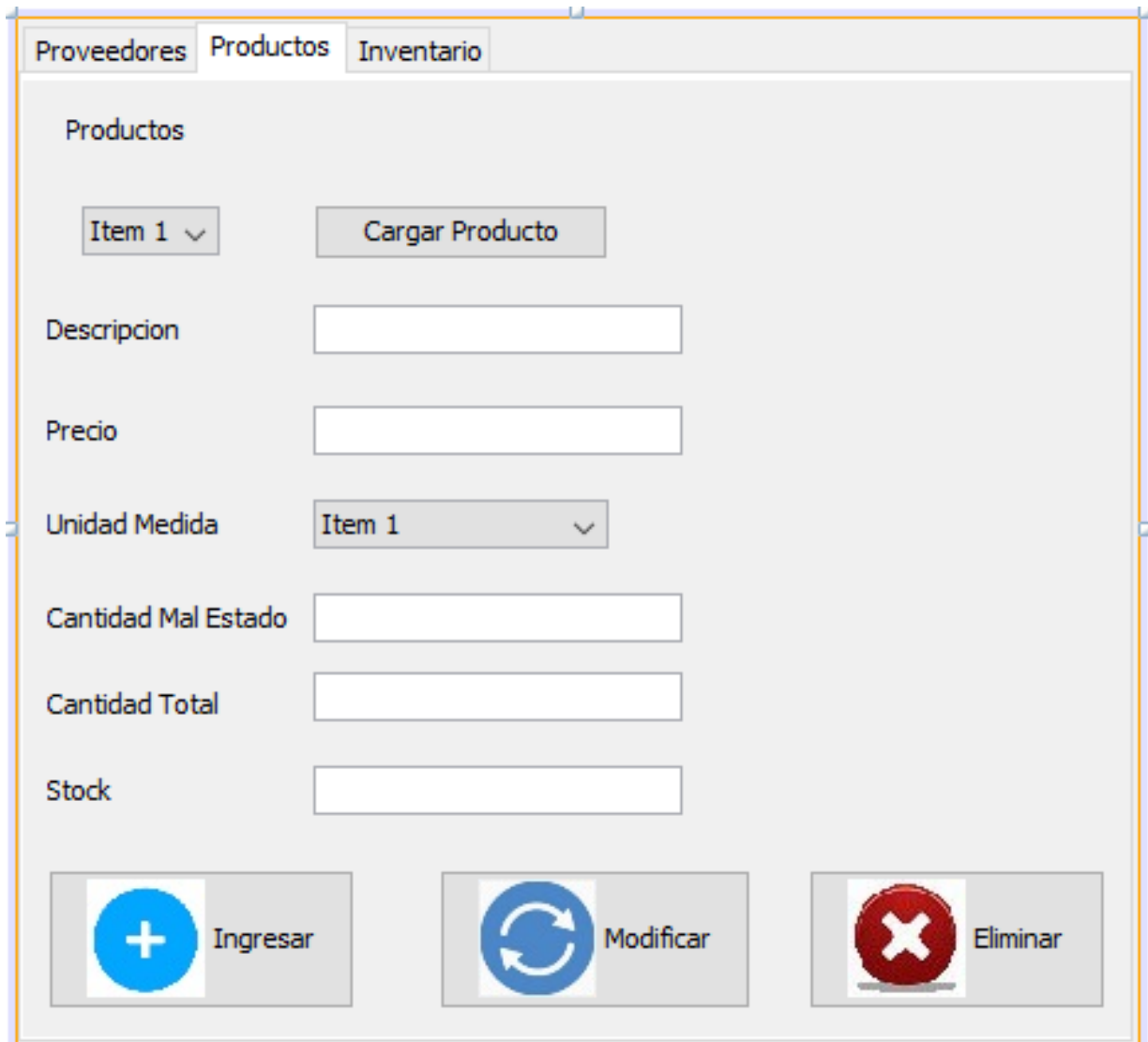
- A dropdown menu labeled 'Item 1' with a downward arrow.
- A button labeled 'Cargar Proveedor'.
- Four text input fields labeled 'Nombre', 'Dirección', 'Teléfono', and 'E-mail'.
- Three action buttons at the bottom: 'Ingresar' (with a blue plus icon), 'Modificar' (with a blue circular refresh icon), and 'Eliminar' (with a red circular delete icon).

Fuente: Jorge Cunalata & David Morán

Pantalla Productos

Esta pantalla es para la creación, modificación, eliminación y consulta de los productos, se autogenera el id del proveedor, se ingresa la descripción, precio, unidad de medida, cantidad en mal estado, cantidad total y su stock.

Ilustración 25: Interfaz Productos



The screenshot displays a web application interface with three tabs: 'Proveedores', 'Productos', and 'Inventario'. The 'Productos' tab is active. The interface includes a dropdown menu labeled 'Item 1' with a downward arrow, and a button labeled 'Cargar Producto'. Below these are several input fields: 'Descripcion', 'Precio', 'Unidad Medida' (with a dropdown menu showing 'Item 1'), 'Cantidad Mal Estado', 'Cantidad Total', and 'Stock'. At the bottom, there are three buttons: 'Ingresar' (with a blue plus icon), 'Modificar' (with a blue circular refresh icon), and 'Eliminar' (with a red X icon).

Fuente: Jorge Cunalata & David Morán

Pantalla Inventario

Esta pantalla es para la creación y vista de los inventarios y los reportes de ventas, se autogenera el id del proveedor y la fecha actual, en donde se genera un archivo de lectura para revisar esta información.

Ilustración 26: Interfaz Inventario

The screenshot displays a web interface with three tabs: 'Proveedores', 'Productos', and 'Inventario'. The 'Inventario' tab is active. Below the tabs, there are two rows of controls. The first row contains a dropdown menu with 'Item 1' selected, a 'Cargar Inventario' button, and a 'Crear Inventario' button. The second row contains another dropdown menu with 'Item 1' selected, a 'Cargar Reporte' button, and the 'Crear Inventario' button. The interface is enclosed in a light gray border with small square handles at the corners.

Fuente: Jorge Cunalata & David Morán

Pantalla Reserva de Mesas

Esta pantalla es para la creación, modificación, eliminación y consulta de las reservas de mesas, se autogenera el id de la reserva, se ingresa la fecha, hora, cantidad de personas, cédula del cliente, nombre del cliente y teléfono del cliente. Se ingresa la mesa disponible (verde) y cambia su estado a ocupado (rojo).

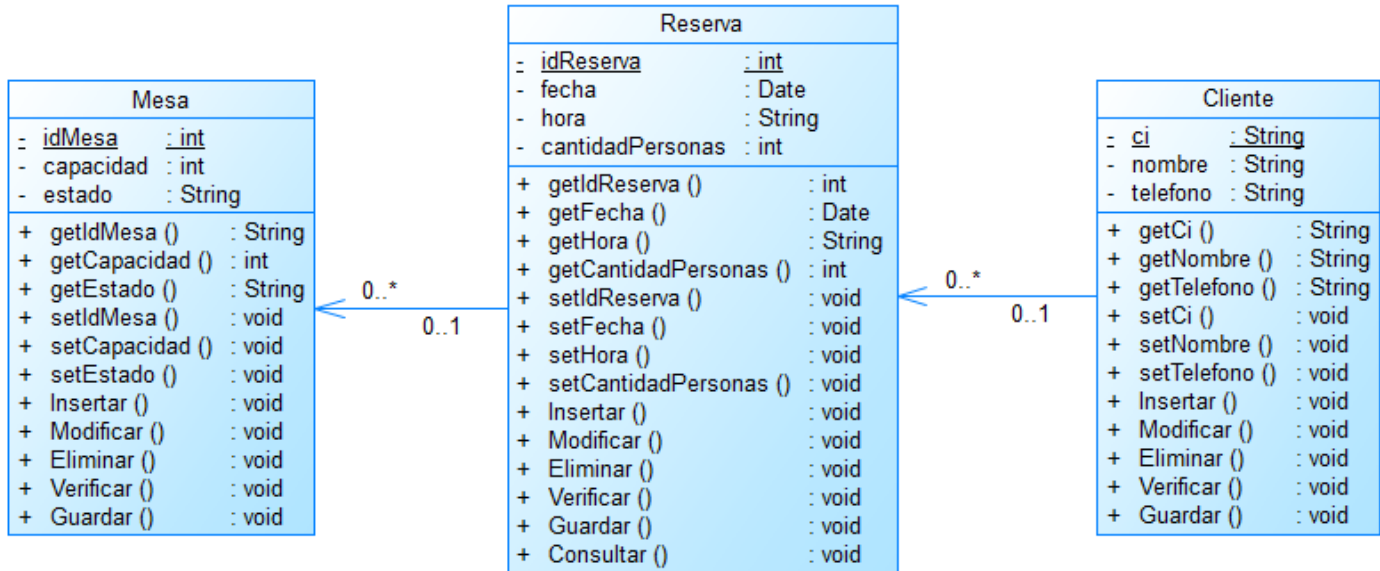
Ilustración 27: Interfaz Reserva de Mesas

The interface is titled "MESAS" and features a form on the left for reservation details and a table of tables on the right. The form includes input fields for "Fecha", "Hora", "Cantidad de personas", "Cédula", "Nombre", and "Teléfono". The table displays four tables, each represented by a square icon with a central square and four legs. The first and third tables are green, indicating they are available, while the second and fourth tables are red, indicating they are occupied. At the bottom of the interface, there are four buttons: "Crear" (blue with a plus sign), "Eliminar" (red with an X), "Modificar" (blue with a circular arrow), and "Consultar" (blue with a magnifying glass).

Fuente: Jorge Cunalata & David Morán

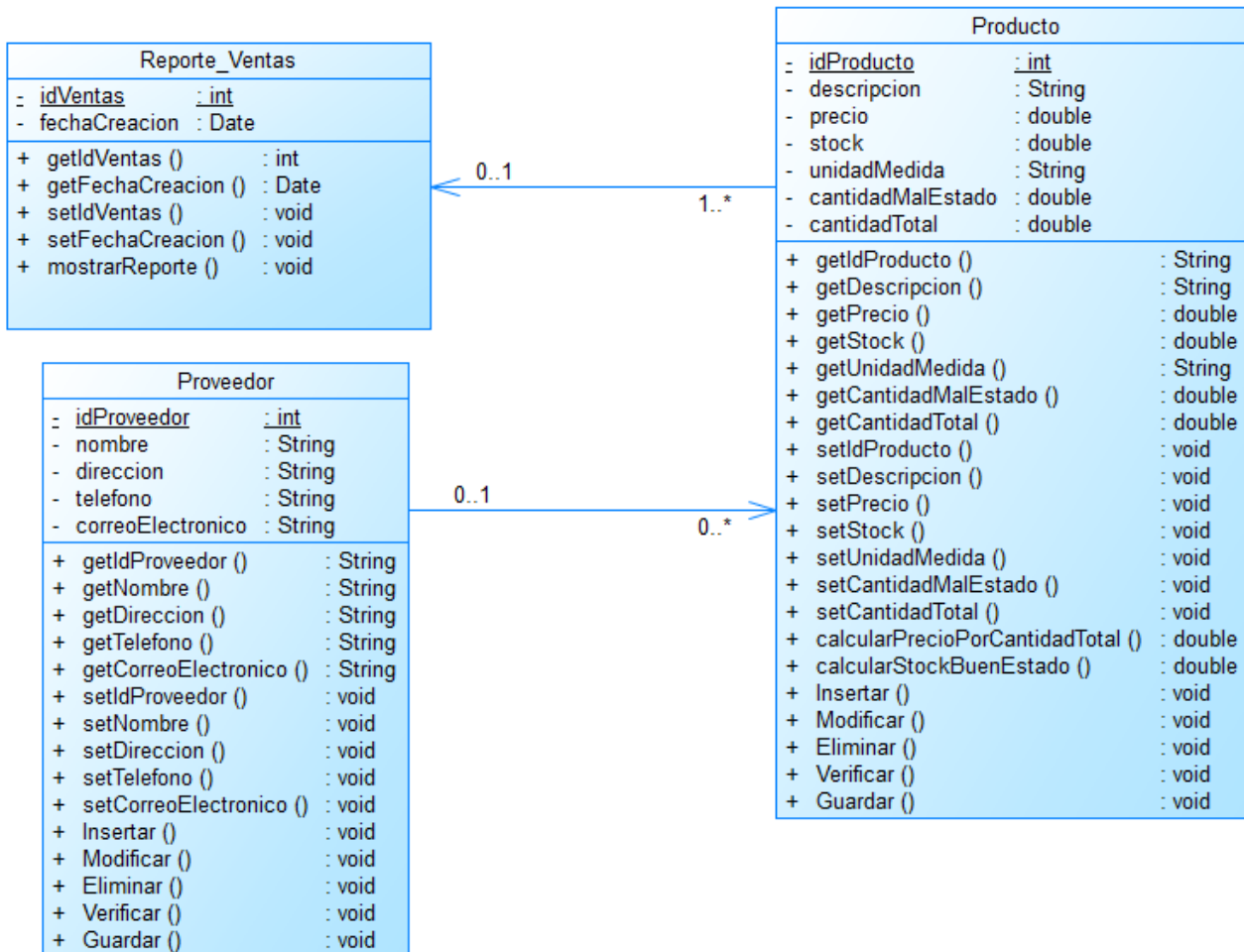
3.1.4 Diagrama de clases

Ilustración 28: Diagrama de Clases Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

Ilustración 29: Diagrama de Clases Proceso de Inventario

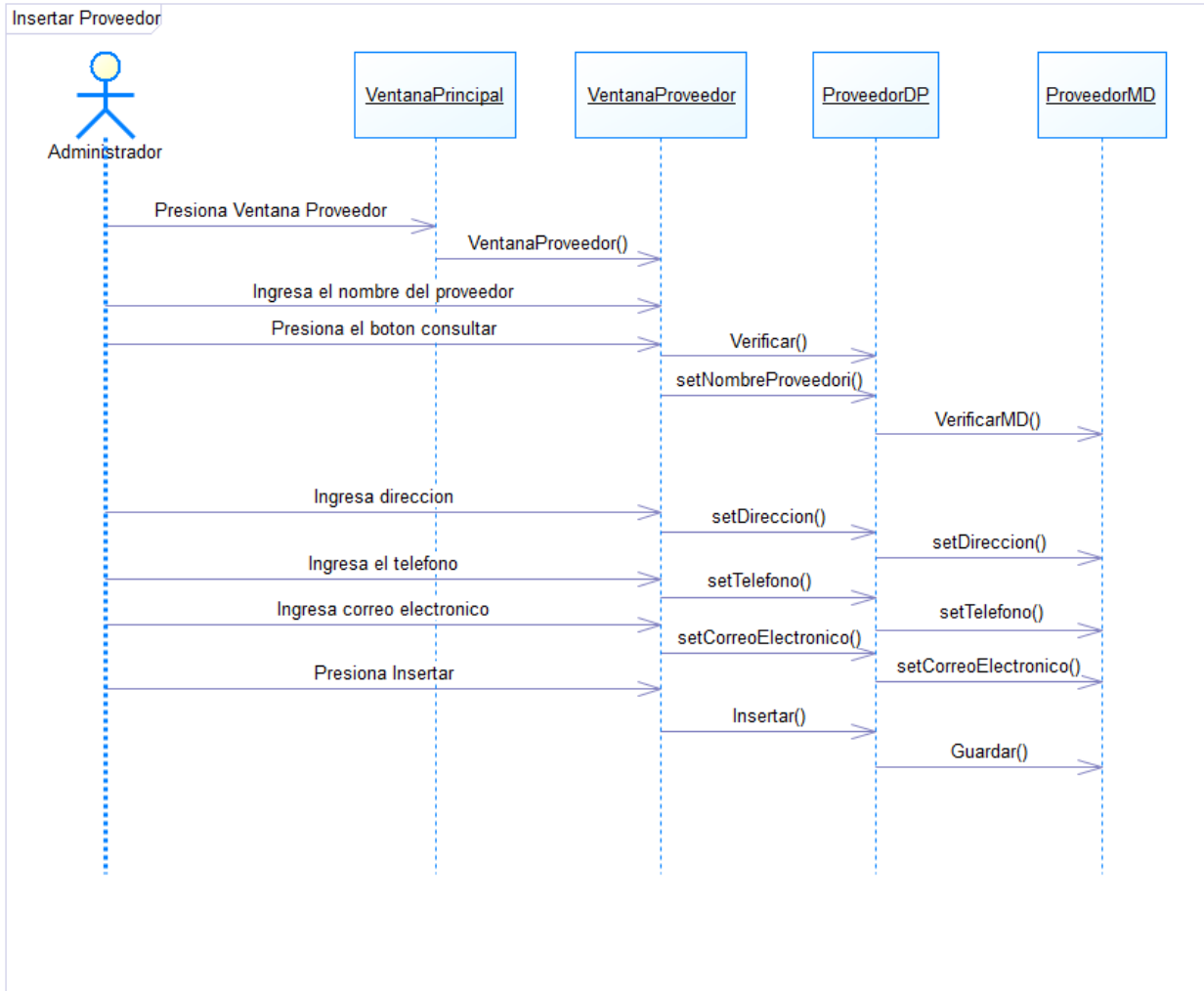


Fuente: Jorge Cunalata & David Morán

3.1.5 Diagrama de secuencia

3.1.5.1 Ingresar Proveedor

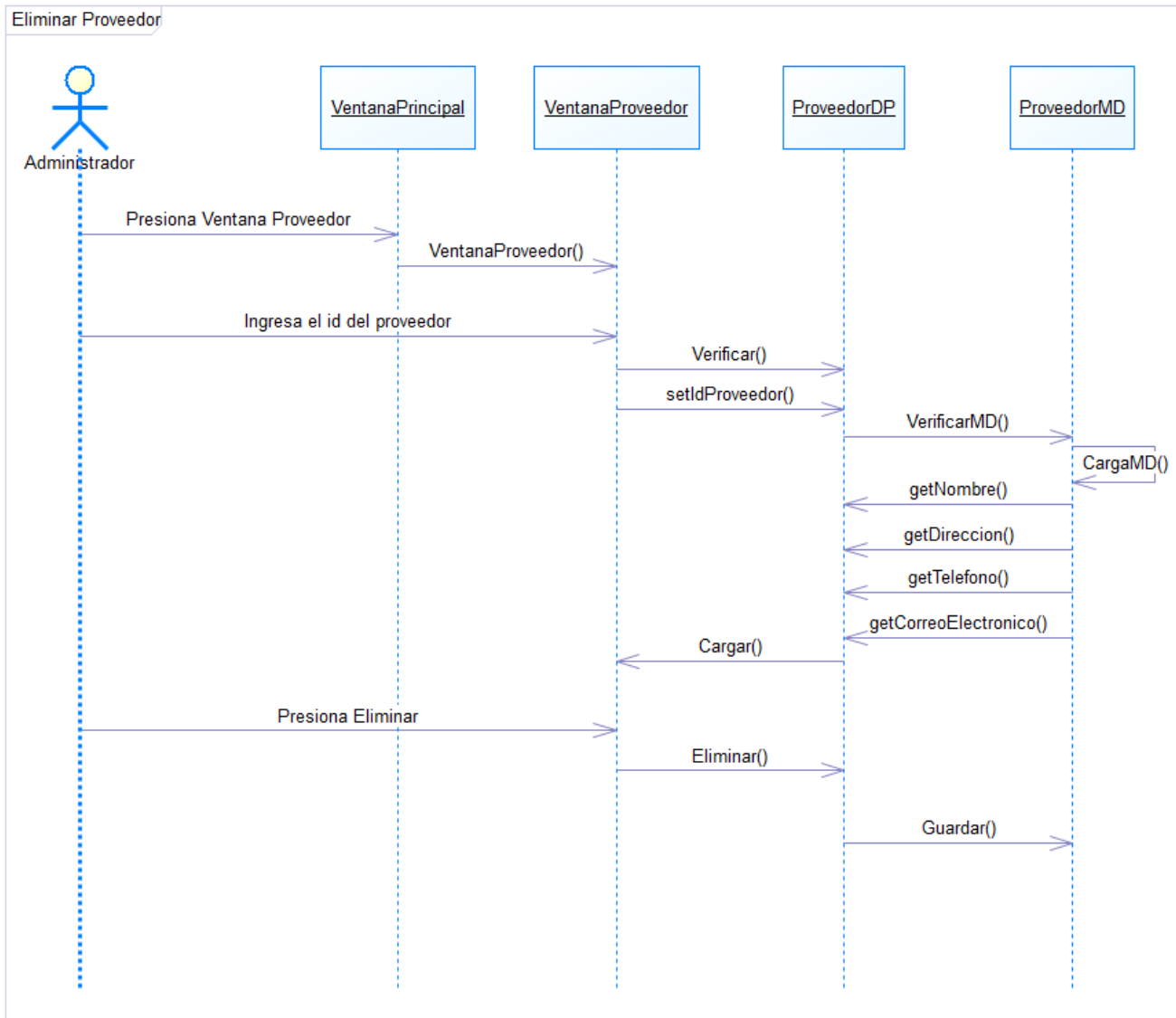
Ilustración 30: Diagrama de Secuencia Insertar Proveedor



Fuente: Jorge Cunalata & David Morán

3.1.5.2 Eliminar proveedor

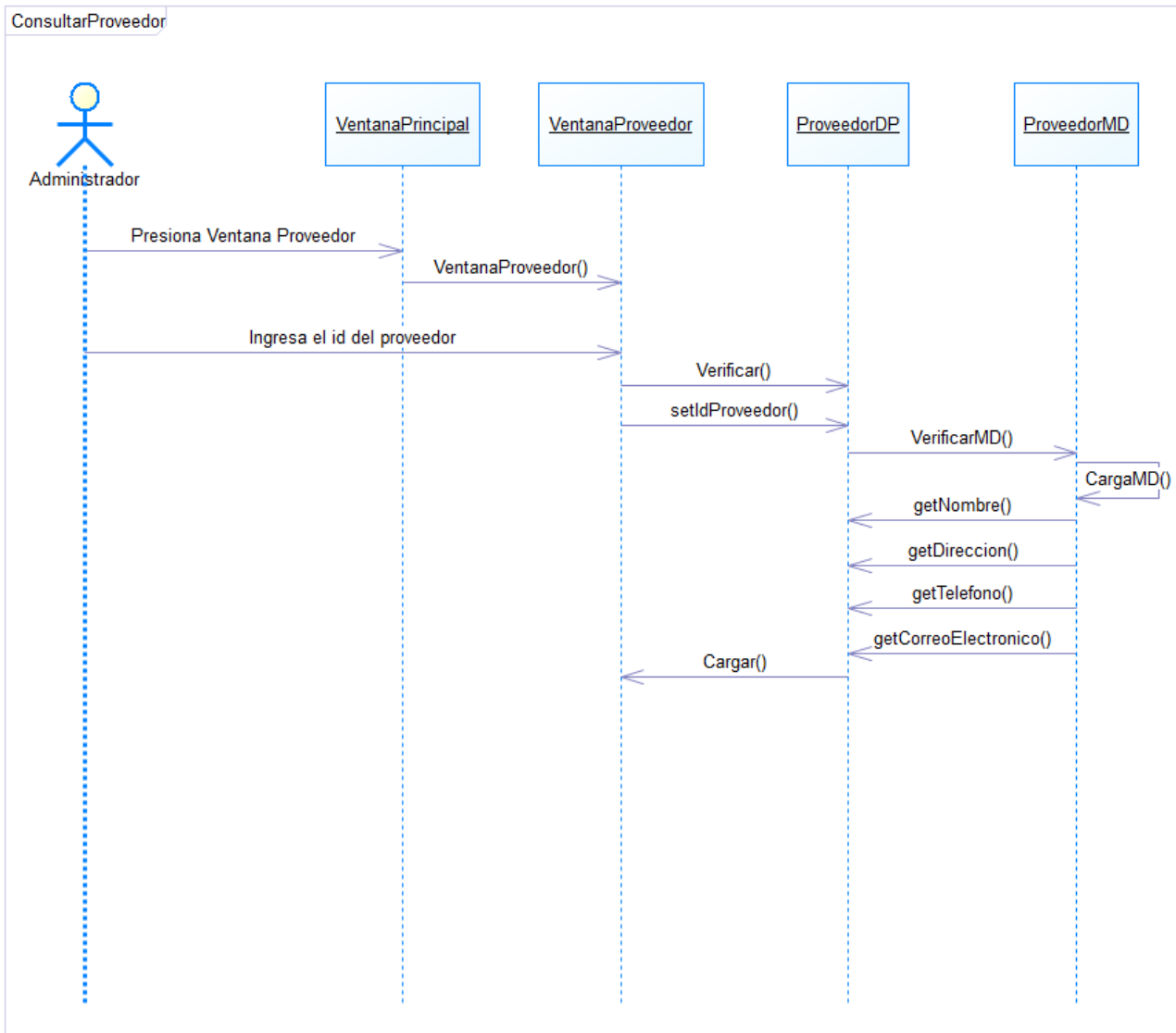
Ilustración 31: Diagrama de Secuencia Eliminar Proveedor



Fuente: Jorge Cunalata & David Morán

3.1.5.3 Consultar proveedor

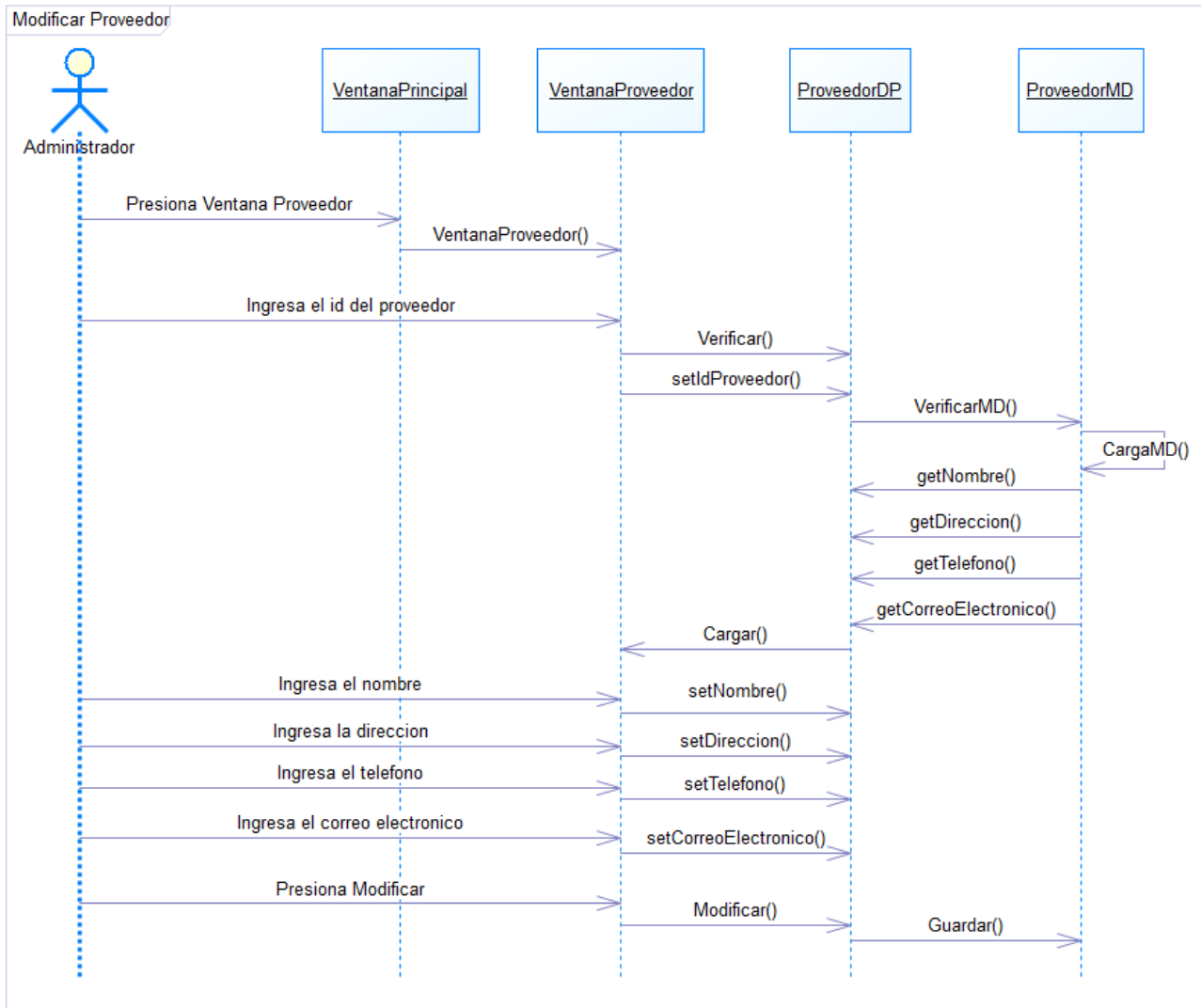
Ilustración 32: Diagrama de Secuencia Consultar Proveedor



Fuente: Jorge Cunalata & David Morán

3.1.5.4 Modificar proveedor

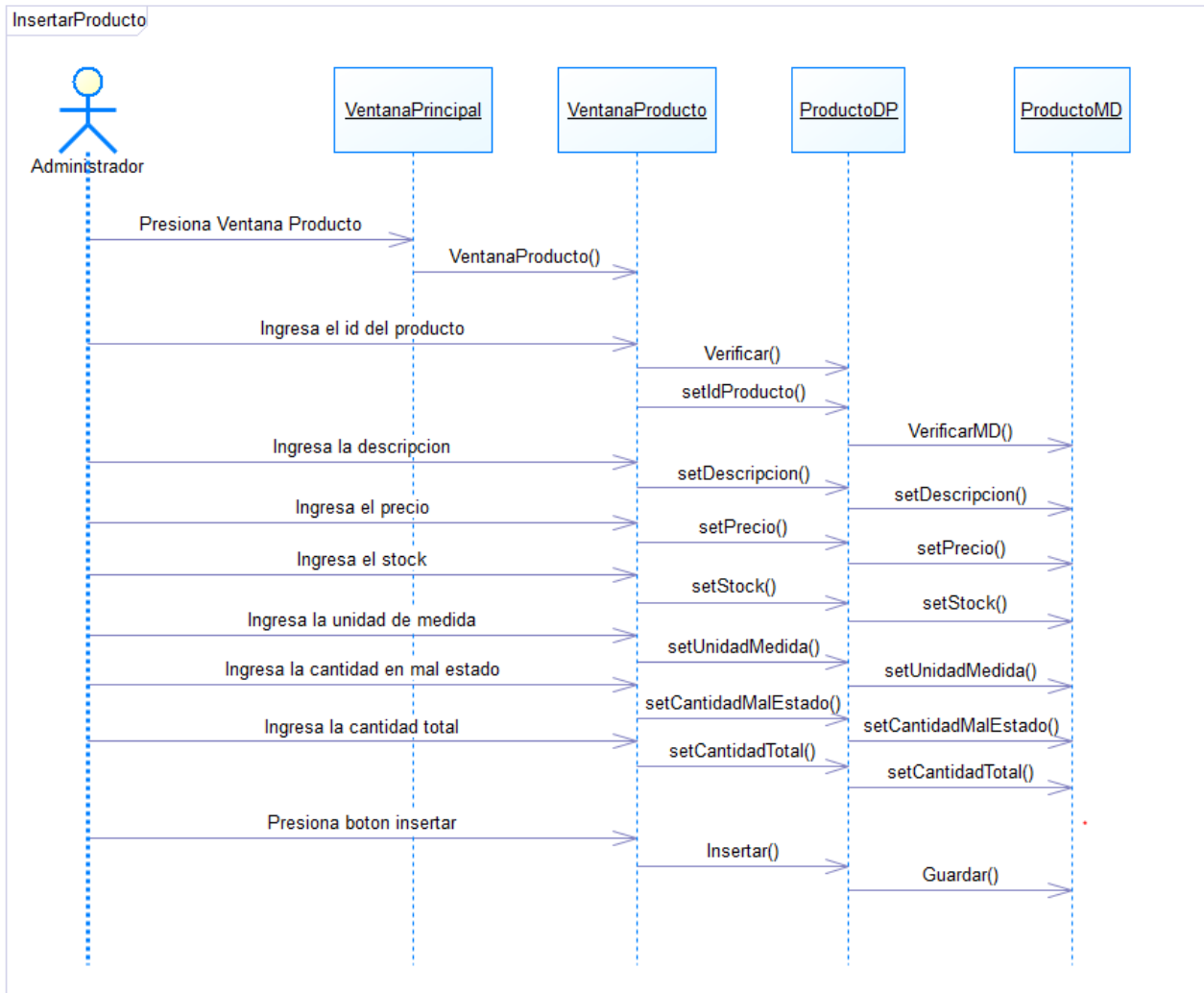
Ilustración 33: Diagrama de Secuencia Modificar Proveedor



Fuente: Jorge Cunalata & David Morán

3.1.1.1 Insertar producto

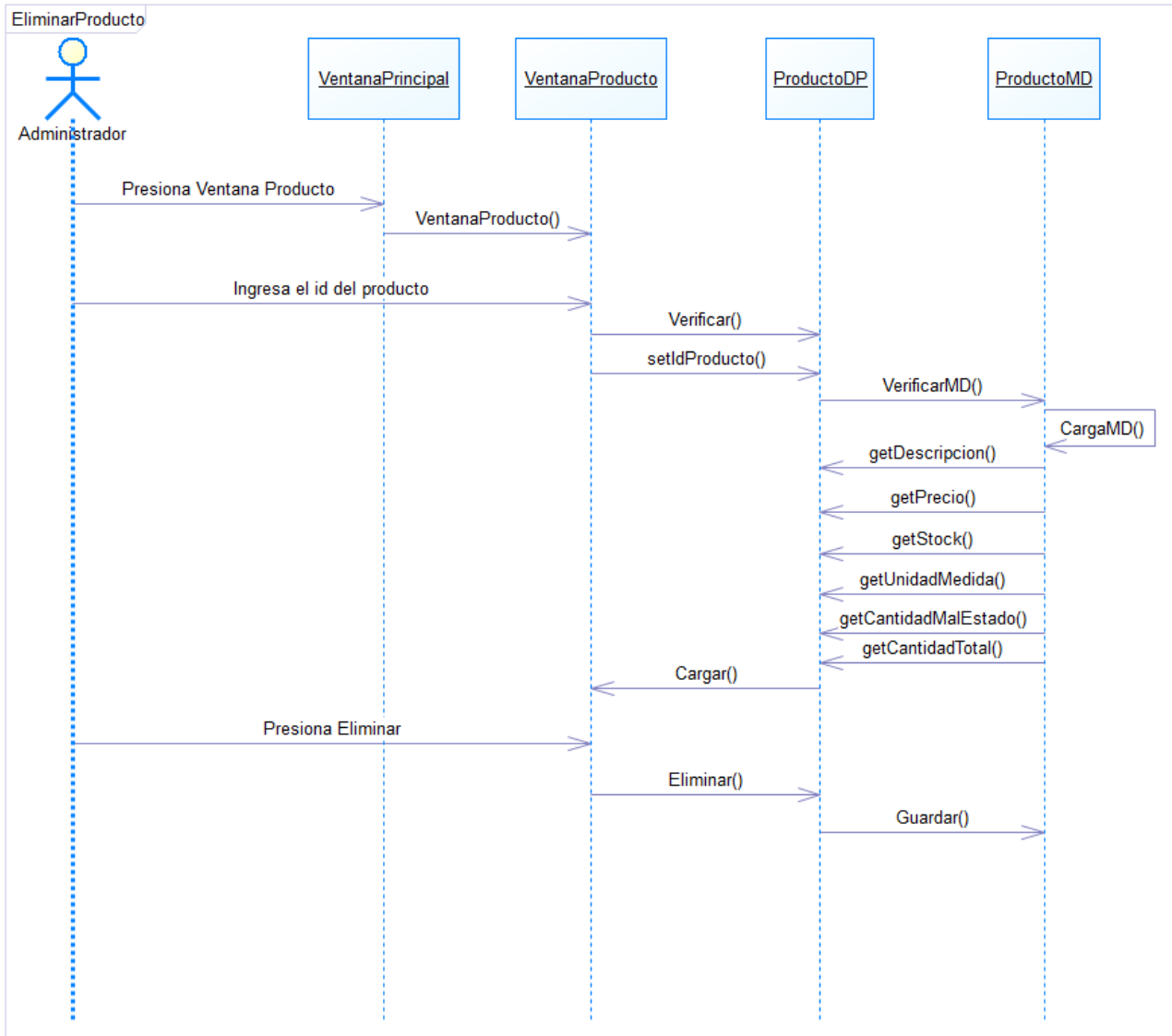
Ilustración 34: Diagrama de Secuencia Insertar Producto



Fuente: Jorge Cunalata & David Morán

3.1.5.5 Eliminar producto

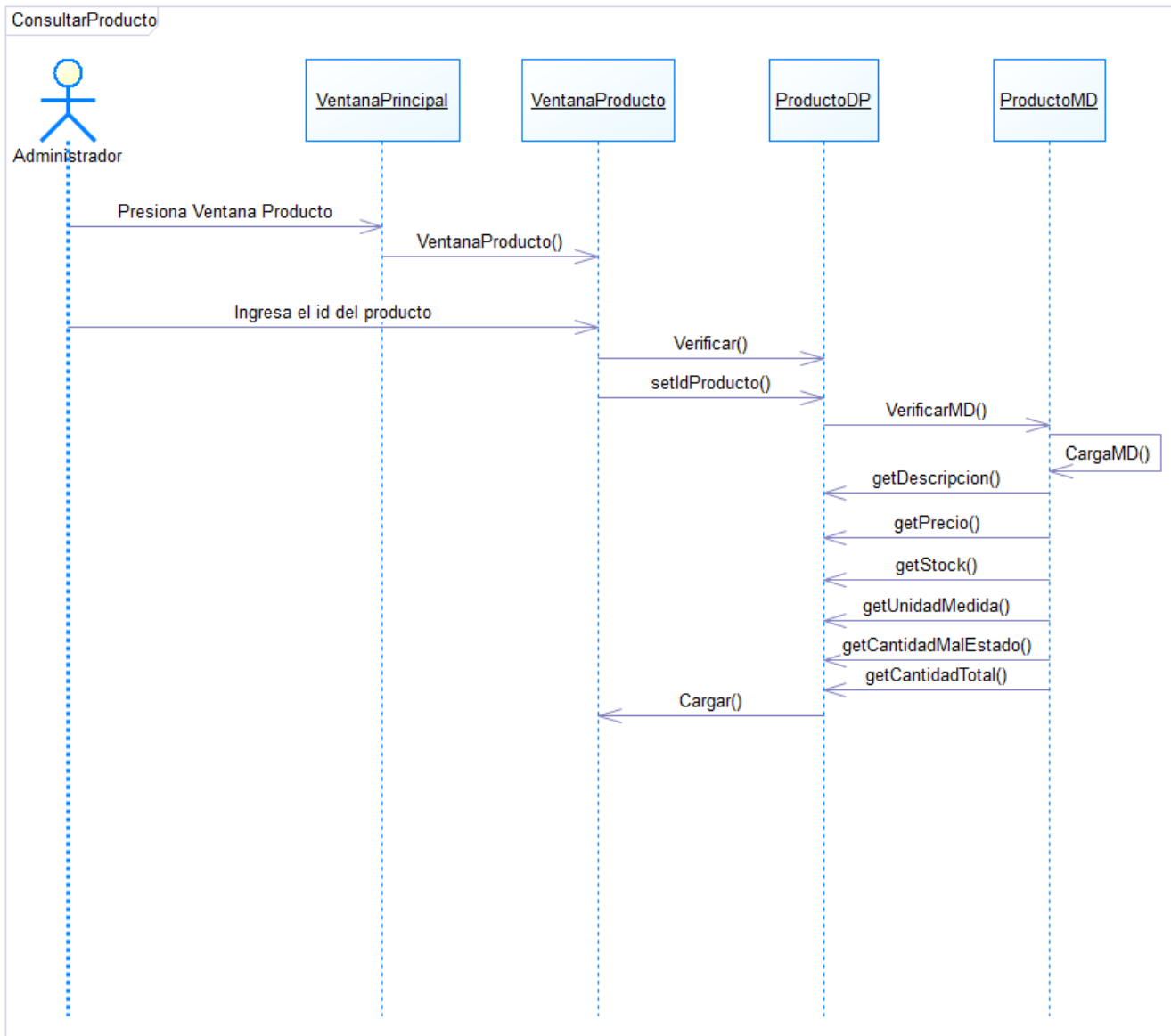
Ilustración 35: Diagrama de Secuencia Eliminar Producto



Fuente: Jorge Cunalata & David Morán

3.1.5.6 Consultar producto

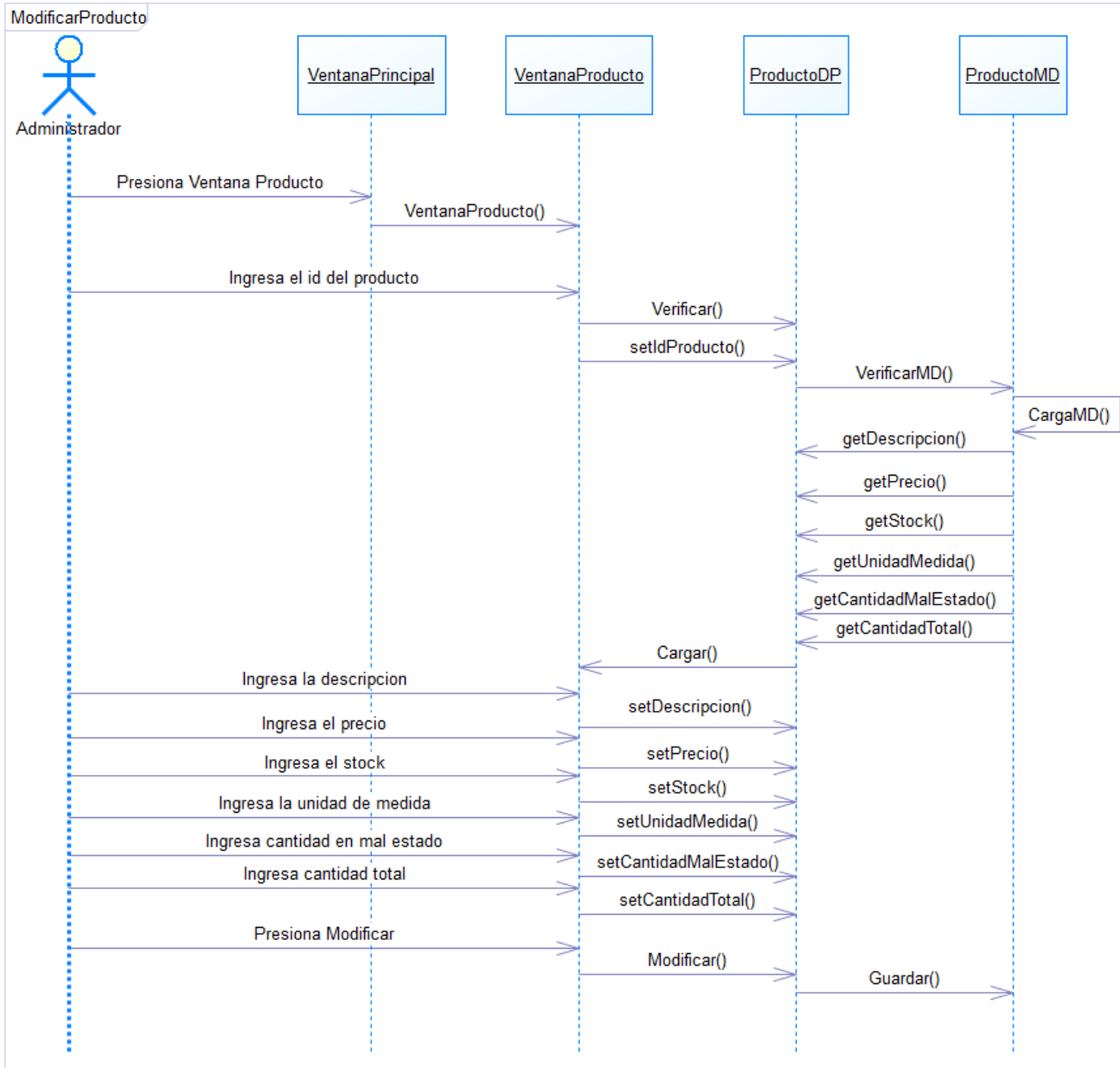
Ilustración 36: Diagrama de Secuencia Consultar Producto



Fuente: Jorge Cunalata & David Morán

3.1.5.7 Modificar producto

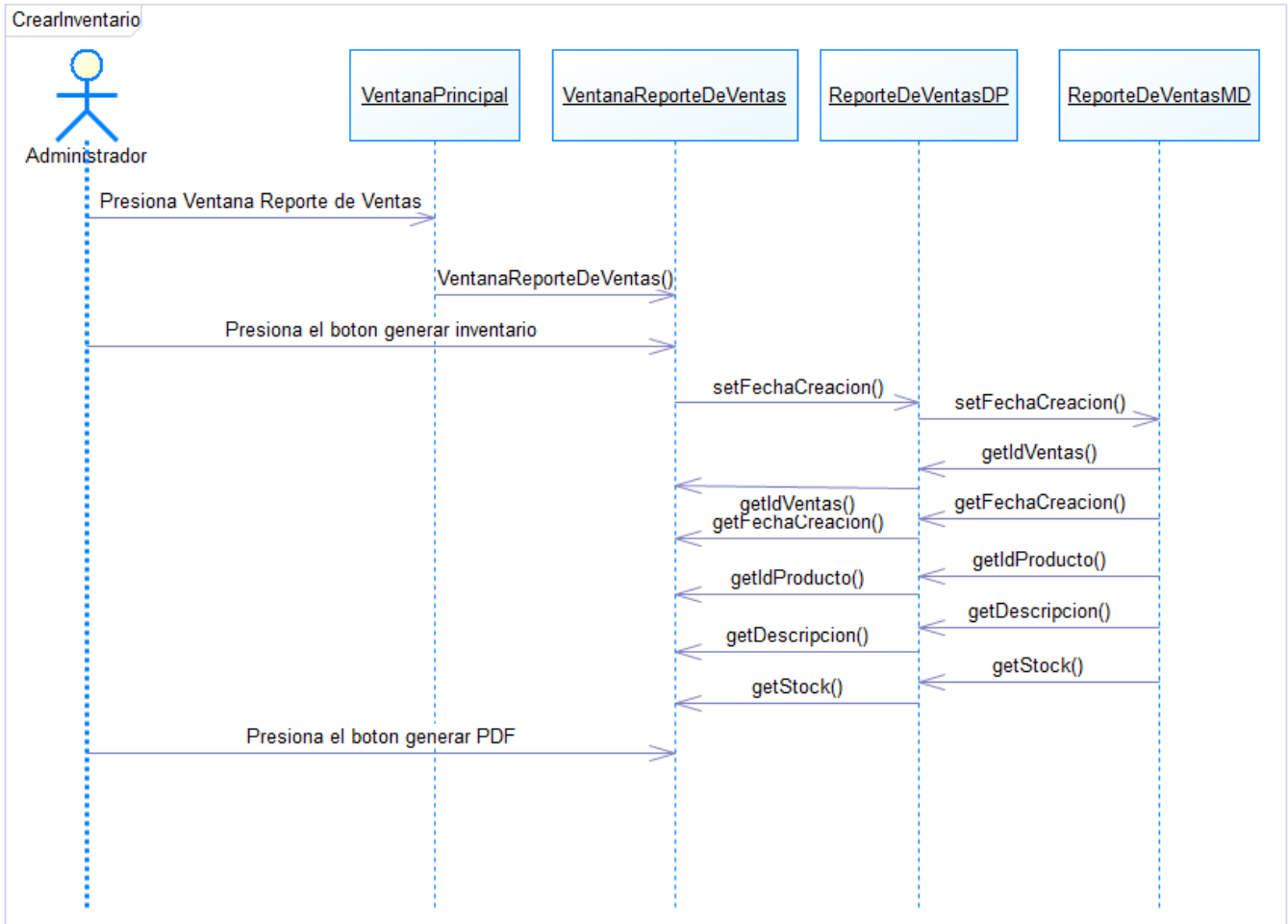
Ilustración 37: Diagrama de Secuencia Modificar Producto



Fuente: Jorge Cunalata & David Morán

3.1.1.1 Crear inventario

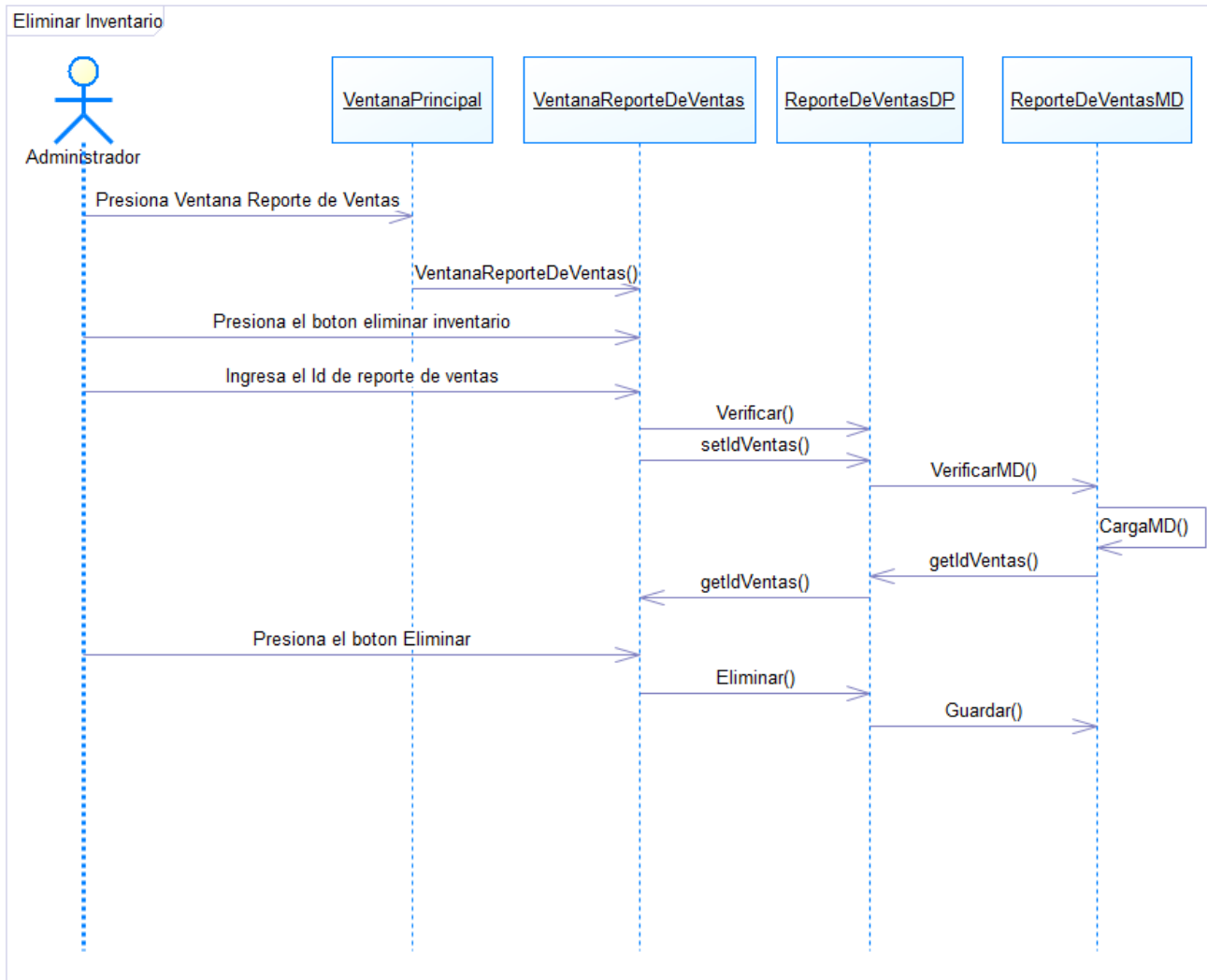
Ilustración 38: Diagrama de Secuencia Crear Inventario



Fuente: Jorge Cunalata & David Morán

3.1.5.8 Eliminar inventario

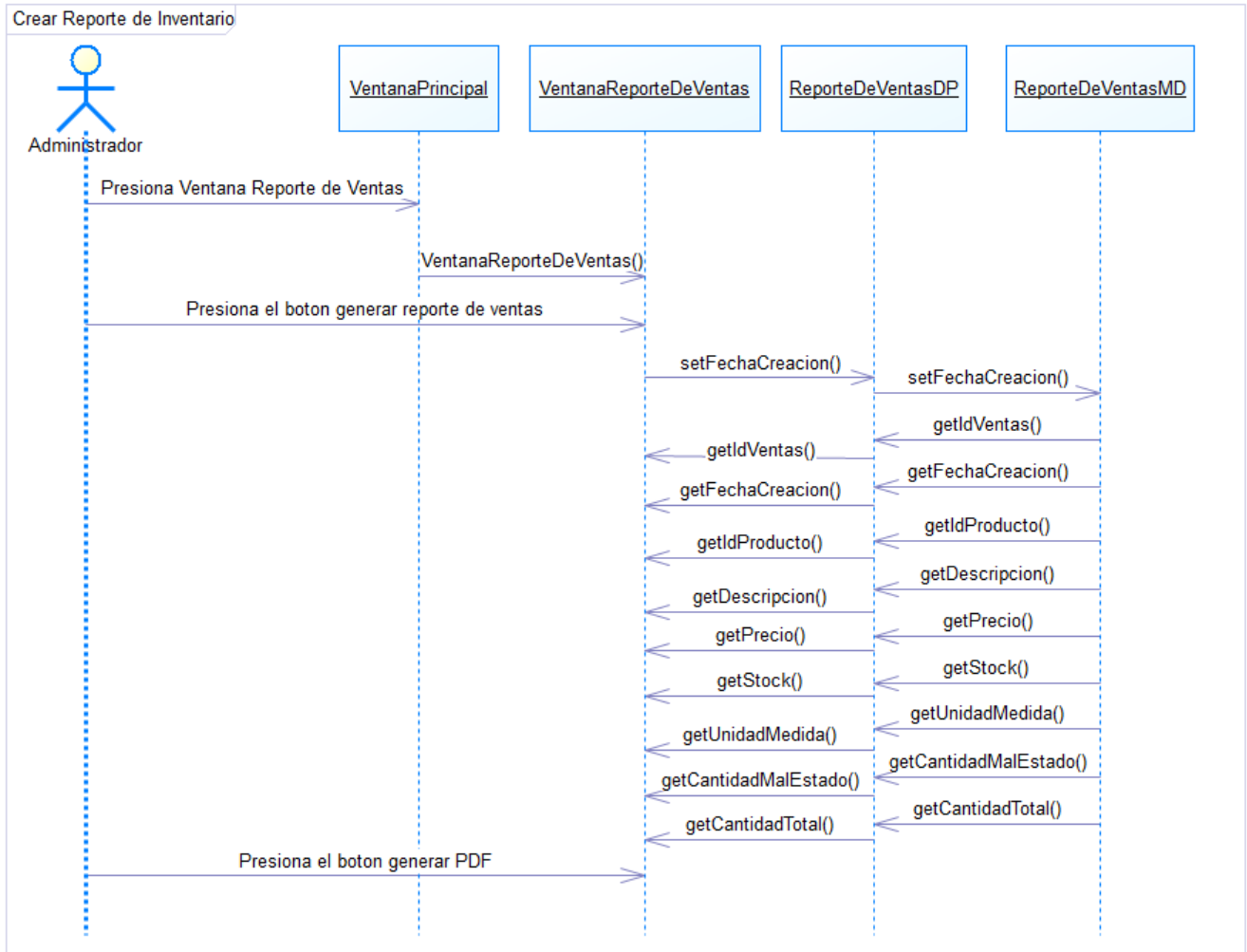
Ilustración 39: Diagrama de Secuencia Eliminar Inventario



Fuente: Jorge Cunalata & David Morán

3.1.5.9 Crear reporte de inventario

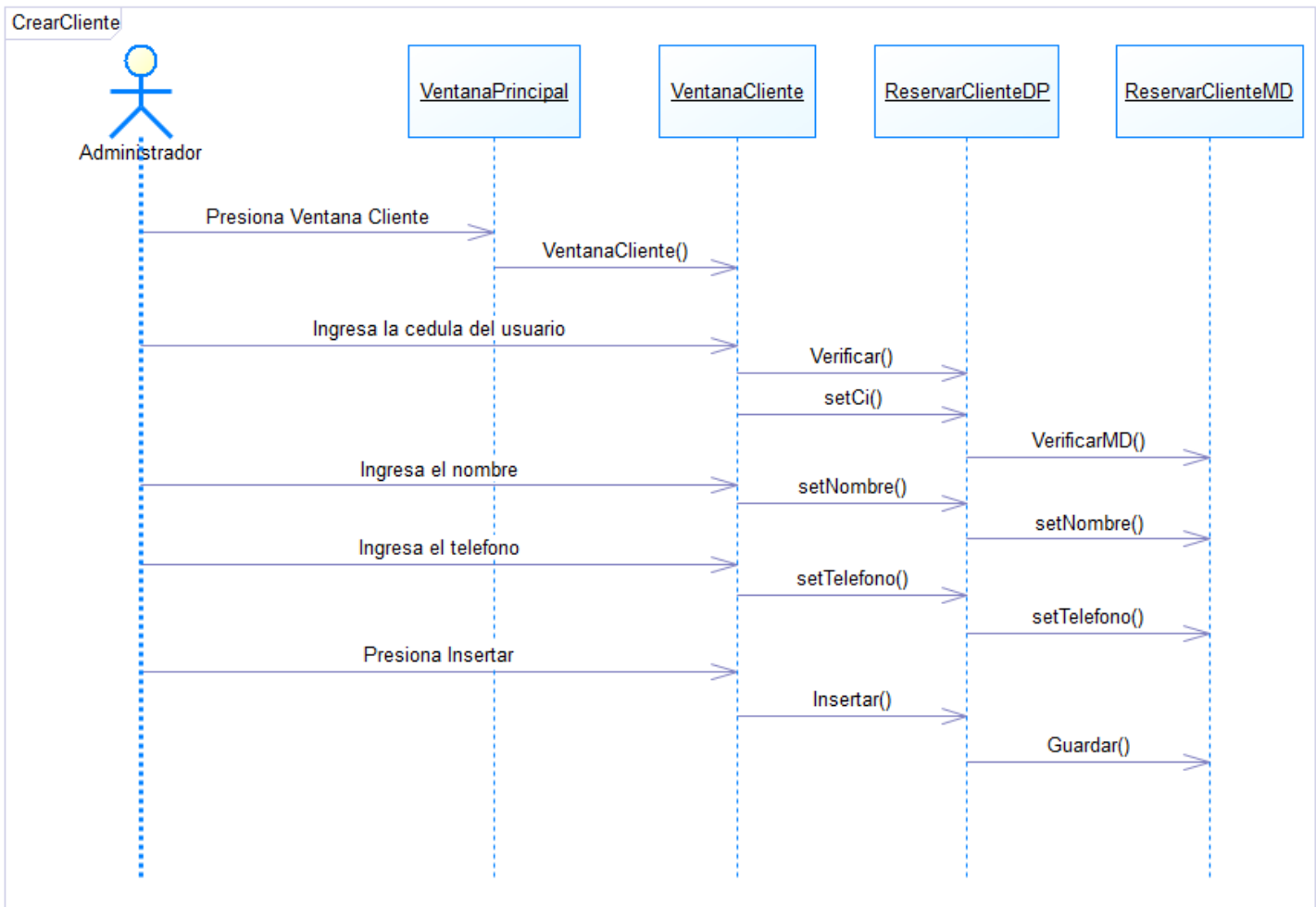
Ilustración 40: Diagrama de Secuencia Crear Reporte de Inventario



Fuente: Jorge Cunalata & David Morán

3.1.5.10 Crear cliente

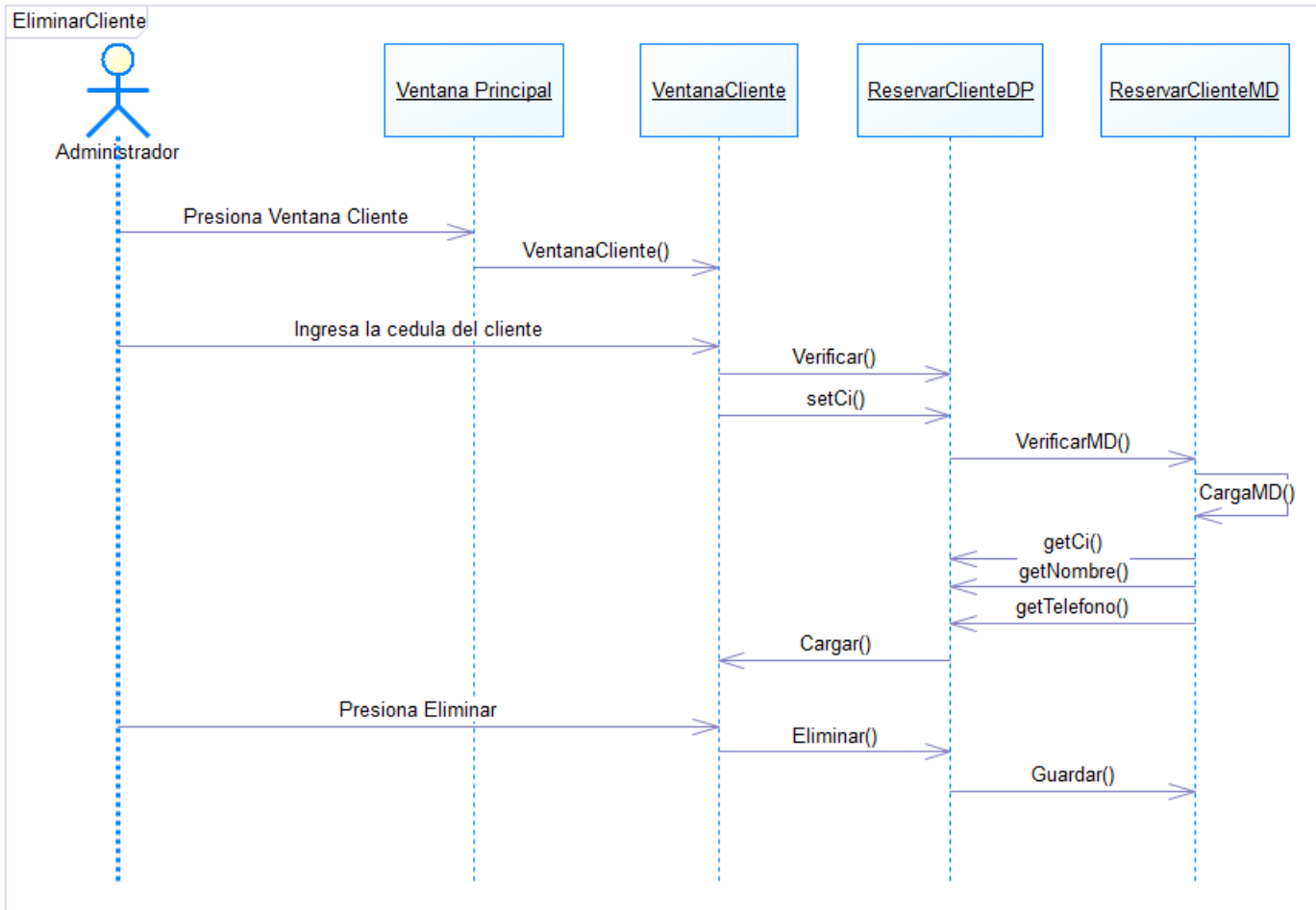
Ilustración 41: Diagrama de Secuencia Crear Cliente



Fuente: Jorge Cunalata & David Morán

3.1.5.11 Eliminar cliente

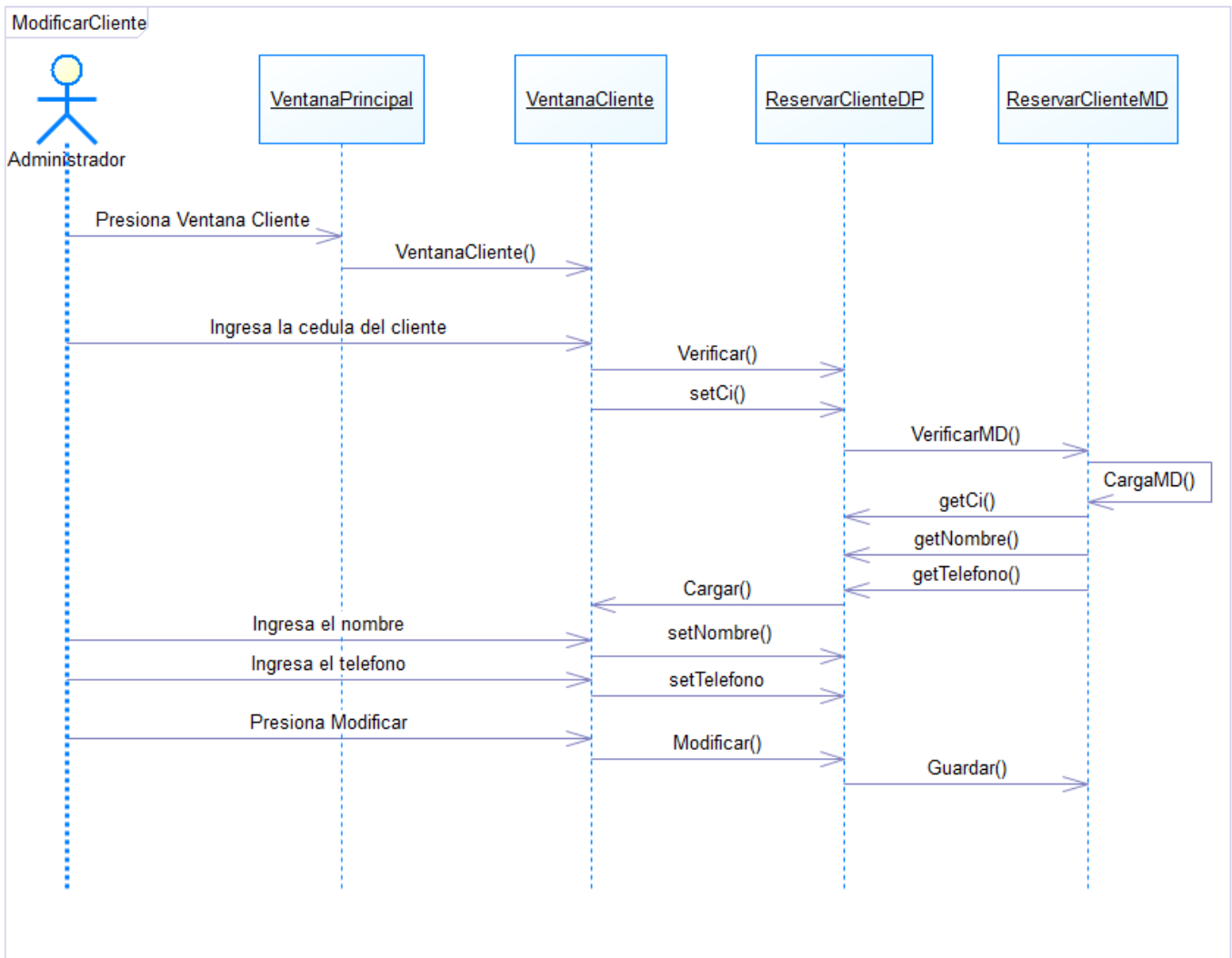
Ilustración 42: Diagrama de Secuencia Eliminar Cliente



Fuente: Jorge Cunalata & David Morán

3.1.5.12 Modificar cliente

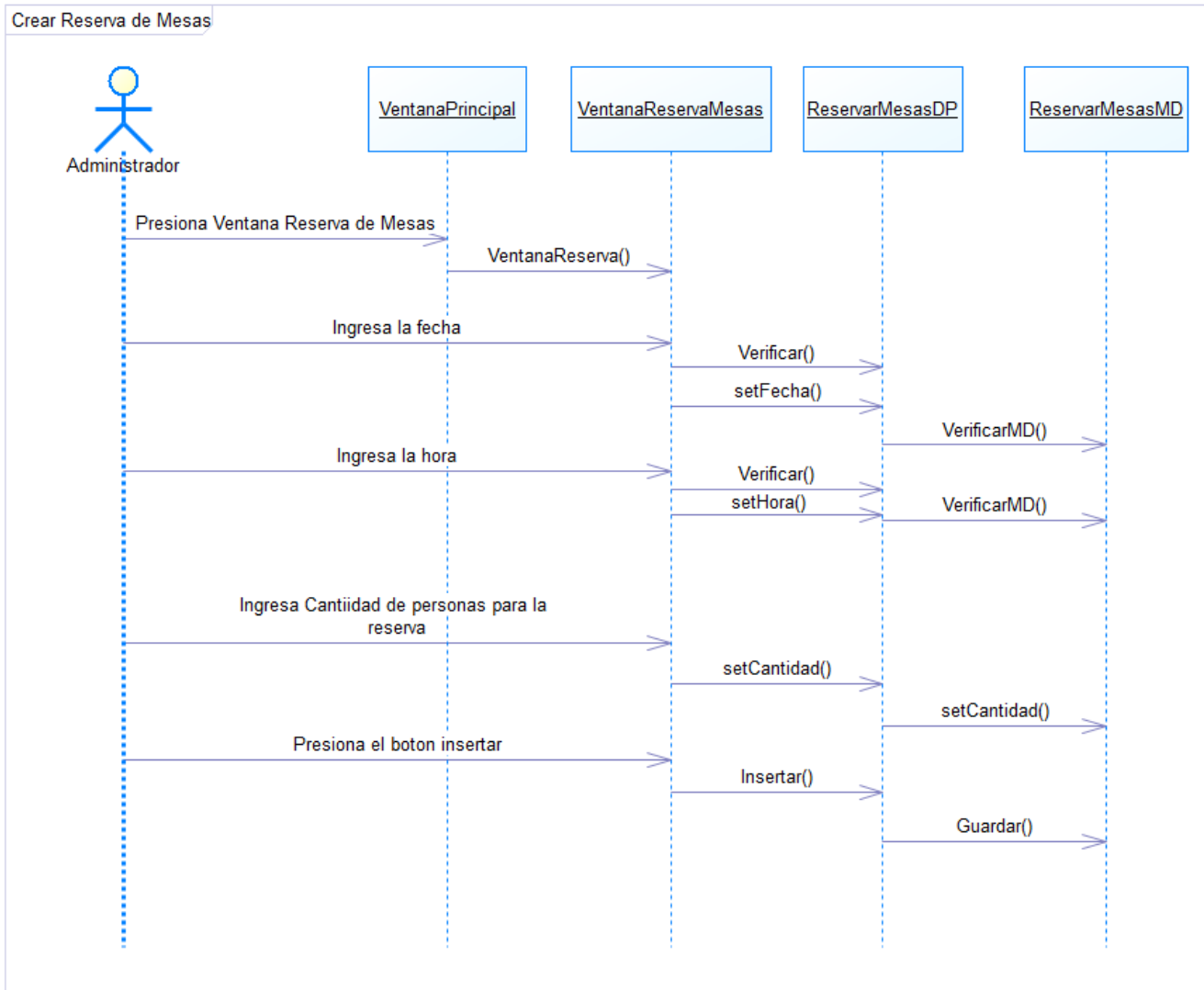
Ilustración 43: Diagrama de Secuencia Modificar Cliente



Fuente: Jorge Cunalata & David Morán

3.1.1.1 Crear reserva de mesas

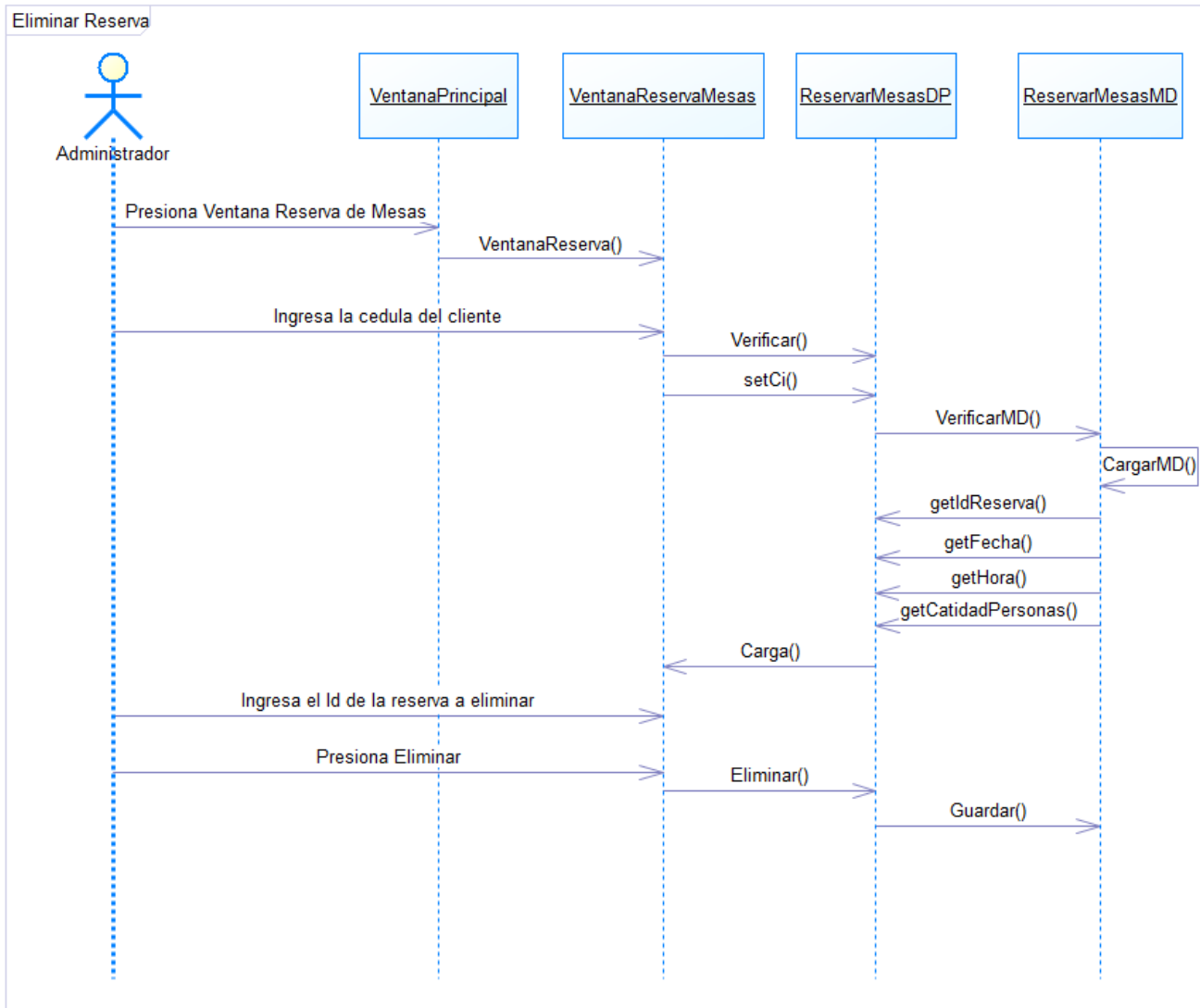
Ilustración 44: Diagrama de Secuencia Crear Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

3.1.5.13 Eliminar reserva de mesas

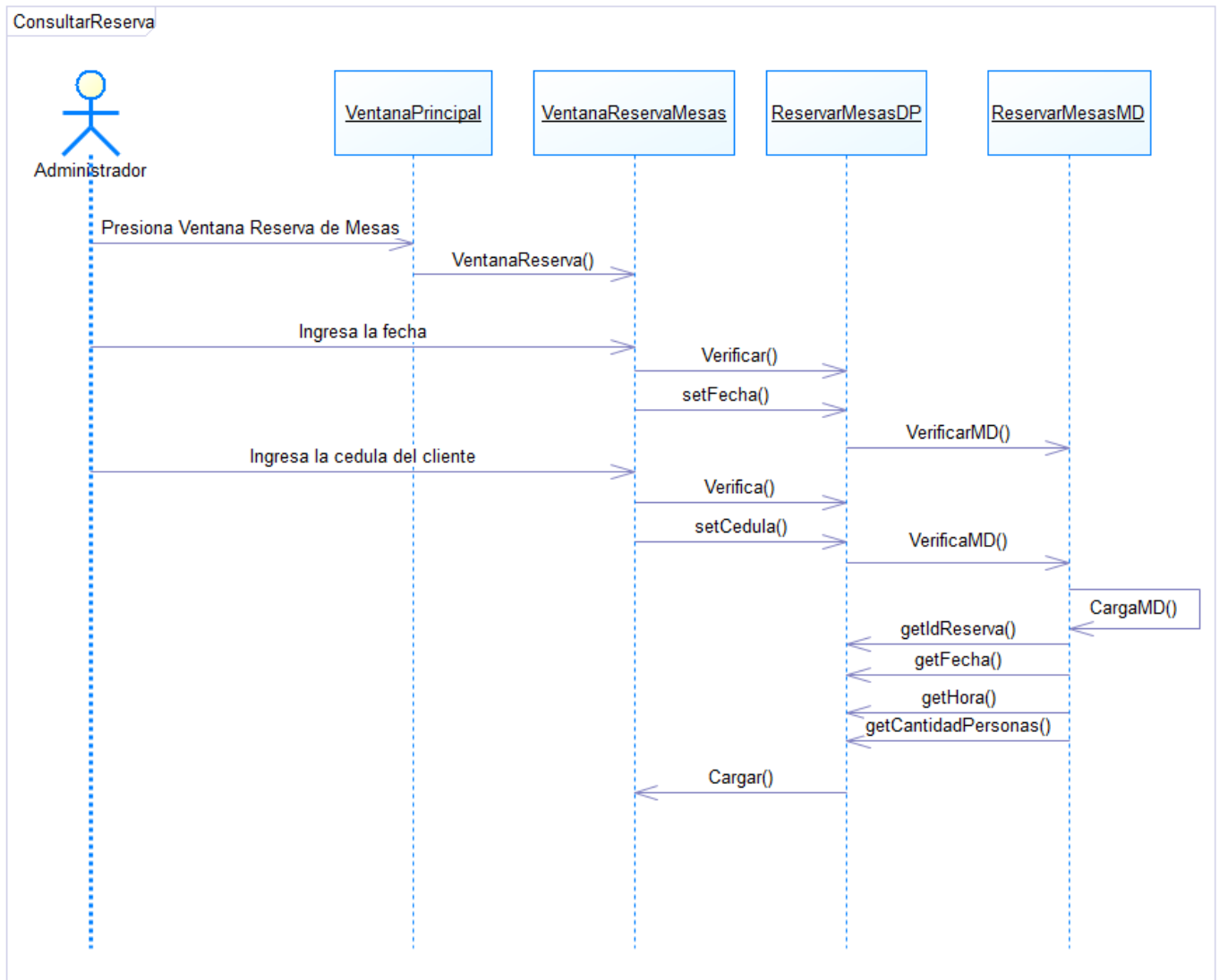
Ilustración 45: Diagrama de Secuencia Eliminar Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

3.1.5.14 Consultar reserva de mesas

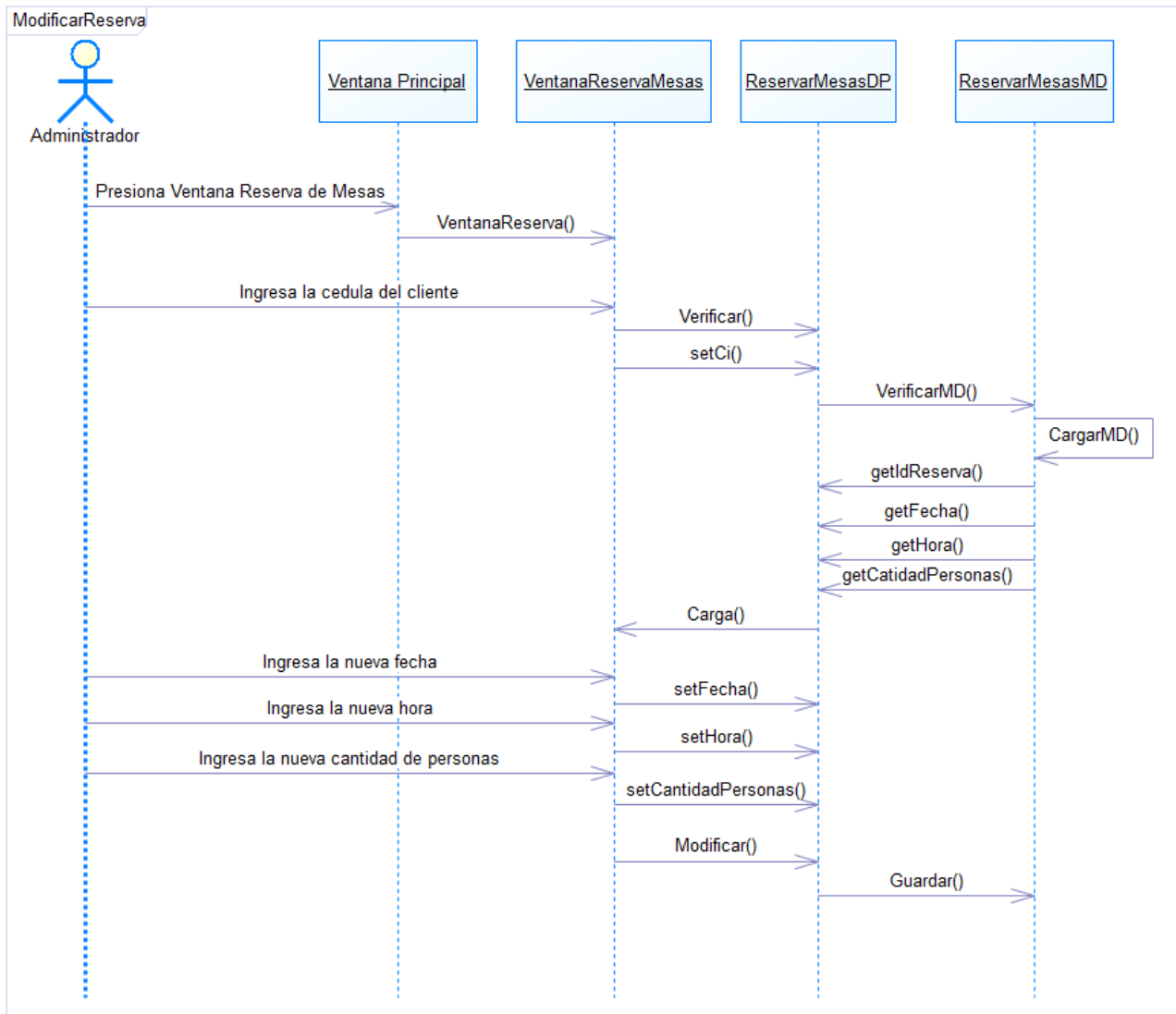
Ilustración 46: Diagrama de Secuencia Consultar Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

3.1.5.15 Modificar reserva de mesas

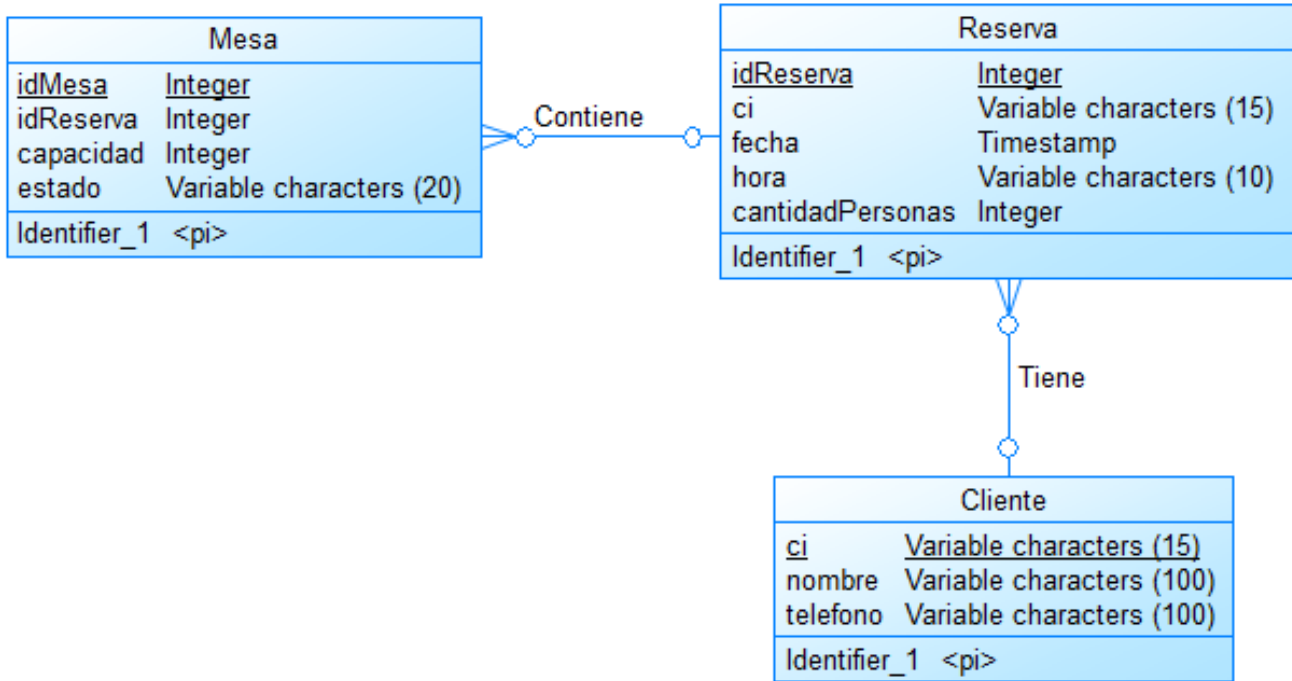
Ilustración 47: Diagrama de Secuencia Modificar Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

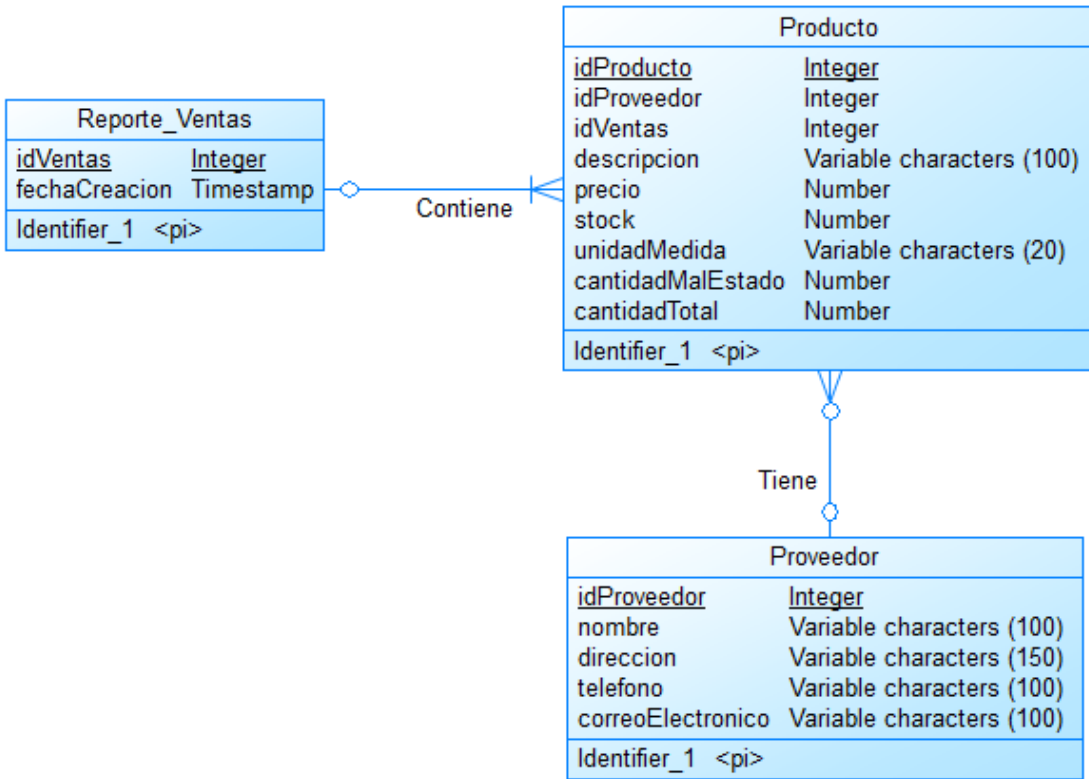
3.1.6 Diagrama entidad relación

Ilustración 48: Diagrama Entidad Relación Reserva de Mesas



Fuente: Jorge Cunalata & David Morán

Ilustración 49: Diagrama Entidad Relación Proceso de Inventario



Fuente: Jorge Cunalata & David Morán

3.1.7 Diccionario de datos

Nombre	Código	Clase	Tipo de Dato	Tamaño	Presición
cantidadMalEstado	CANTIDADMALESTADO	Producto	Double	5	2
cantidadPersonas	CANTIDADPERSONAS	Reserva	integer		
cantidadTotal	CANTIDADTOTAL	Producto	Double	5	2
capacidad	CAPACIDAD	Mesa	integer		
ci	CI	Cliente	String	15	
correoElectronico	CORREOELECTRONICO	Proveedor	String	100	
descripcion	DESCRIPCION	Producto	String	100	
direccion	DIRECCION	Proveedor	String	150	
estado	ESTADO	Mesa	String	20	
fecha	FECHA	Reserva	Date		
fechaCreacion	FECHACREACION	Reporte_Ventas	Date		
hora	HORA	Reserva	String	10	
idMesa	IDMESA	Mesa	integer		
idProducto	IDPRODUCTO	Producto	integer		
idProveedor	IDPROVEEDOR	Proveedor	integer		
idReserva	IDRESERVA	Reserva	integer		
idVentas	IDVENTAS	Reporte_Ventas	integer		
nombre	NOMBRE	Proveedor	String	100	
nombre	NOMBRE	Cliente	String	100	
precio	PRECIO	Producto	Double	5	2
stock	STOCK	Producto	Double	5	2
telefono	TELEFONO	Proveedor	String		
telefono	TELEFONO	Cliente	String		
unidadMedida	UNIDADMEDIDA	Producto	String	20	

Tabla 1: Diccionario de Datos

Fuente: Jorge Cunalata & David Morán

CAPÍTULO 4

4. FASE DE CONSTRUCCIÓN

4.1 Implementación

4.1.1 Estándares de implementación

- Estándares de Implementación de las Clases

El nombre de las clases irá con la primera letra en mayúscula y el resto en minúscula.

- Estándares de Implementación de Paquetes

El nombre de los paquetes irá todo con mayúscula

- Estándares de Implementación de Atributos

El nombre de los atributos irán con escritura camello es decir la primera palabra todo con minúscula y si fuera un atributo compuesto de dos palabras la segunda iría pagada, con su primera letra en mayúscula y el resto en minúscula.

Ejemplo: cantidadPersonas (atributo de la clase de Reserva).

- Estándares de Implementación de las Funciones

El nombre de las funciones irán con escritura camello es decir la primera palabra todo con minúscula y si fuera una función compuesto de dos palabras la segunda iría pagada, con su primera letra en mayúscula y el resto en minúscula.

Ejemplo: setCandidadPersonas (función de la clase de Reserva).

4.1.2 Definición de clases

Clase Proveedor

Esta clase contiene todos los atributos del proveedor los cuales son: idProveedor, nombre, direccion, telefono, correoElectronico.

Clase Producto

Esta clase contiene todos los atributos de un producto los cuales son: idProducto, descripcion, precio, stock, unidadMedida, cantidadMalEstado, cantidadTotal.

Clase Reporte_Ventas

Esta clase tiene todos los atributos para la creación de un reporte de ventas los que son: idVentas, fechaCreacion.

Clase Mesa

Esta clase tiene todos los atributos de una mesa para su reservación los cuales son: idMesa, capacidad, estado.

Clase Cliente

Esta clase tiene todos los atributos para de un cliente necesarios para la reservación de una mesa los cuales son: ci, nombre, telefono.

Clase Reserva

Esta clase tiene todos los atributos para la creación de una reserva de mesa los cuales son: idReserva, fecha, hora, cantidadPersonas.

4.1.3 Codificación de clases

Las clases necesarias a codificarse son las siguientes:

- Clase Proveedor
- Clase Producto
- Clase Reporte_Ventas
- Clase Mesa
- Clase Cliente
- Clase Reserva

El código fuente de estas clases se encuentran en el ANEXO A

4.1.4 Conexión con base de datos

LA conexión a la base de datos se realizará mediante un conector Postgres jdbc y cuyo código de implementación se encuentra en el ANEXO B.

4.1.5 Manual técnico

El manual técnico se encuentra en el ANEXO C.

4.1.6 Manual de usuario

El manual de usuario se encuentra en el ANEXO D.

4.1.7 Código fuente

El código fuente se encuentra en el ANEXO E.

4.2 Pruebas

4.2.1 Plan de pruebas del sistema

Mediante una definición de plan de pruebas para el sistema se intenta encontrar la mayor cantidad de defectos que contenga el programa desarrollado para así corregirlo. Estas

pruebas están pensadas funcional y técnicamente para un correcto desempeño del sistema frente a las necesidades del usuario final.

Tabla 2: Plan de Pruebas del Sistema – Gestión de Reserva de Mesas

Caso de Prueba		GESTIÓN DE RESERVA DE MESAS	
Descripción		La prueba de Gestión de Reserva de Mesas intenta probar la calidad del requerimiento de administrar la reserva de mesas tanto como clientes, mesas y las reservas.	
Precondiciones		Que exista el registro de las mesas precargadas en la base de datos con su id, capacidad y su estado que por defecto será disponible.	
FUNCIONALIDAD	ENTRADA	RESULTADO ESPERADO	CUMPLE
Ingresar clientes para la reserva de mesas	Ingresar: ci (número de cédula o pasaporte), nombre (nombre del cliente) y telefono (teléfono del cliente).	En el caso de que se ingresen bien los datos se muestra un mensaje de Datos Correctamente Ingresados , en el caso de que no sean los datos correctos un mensaje de ERROR	SI
Modificar cliente para una reserva de mesa	Ingresar los nuevos datos de: ci (número de cédula o pasaporte), nombre (nombre del cliente) y telefono (teléfono del cliente).	En el caso de que se modifiquen bien los datos se muestra un mensaje de Datos Correctamente Modificados , en el caso de que no sean los datos correctos un mensaje de ERROR	SI
Eliminar un cliente	Se selecciona el cliente y se presiona el botón eliminar	En caso de ser eliminado correctamente se presenta un mensaje Eliminado	SI

		Exitosamente en otro caso un mensaje de ERROR.	
Crear una reserva de mesa	Se ingresa a la pantalla de reserva de mesas y se ingresan los datos de la reserva que son: fecha, hora y cantidadPersonas y se presiona el botón Crear	Se muestra un mensaje de Exitosamente Creada en caso de hacerlo caso contrario se presenta un mensaje de ERROR	SI
Eliminar una reserva de mesa	Se ingresa a la pantalla de reserva de mesas y se selecciona la reserva y se presiona el botón eliminar.	En caso de ser eliminado correctamente se presenta un mensaje Eliminado Exitosamente en otro caso un mensaje de ERROR.	SI
Consultar una reserva de mesa	Se ingresa en la pantalla reserva de mesas y se ingresan la fecha y la cédula del cliente se presiona el botón Consultar	Se presentan los datos de la reserva consultada en caso de existir o sinó se presenta un mensaje No Existe Reserva	SI
Modificar una reserva de mesa	Se ingresa en la pantalla reserva de mesas y se ingresan la fecha y la cédula del cliente se presiona el botón Consultar, se modifican los campos con la nueva información y se presiona el botón Modificar	En el caso de que se modifiquen bien los datos se muestra un mensaje de Datos Correctamente Modificados, en el caso de que no sean los datos correctos un mensaje de ERROR	SI

Fuente: Jorge Cunalata & David Morán

Tabla 3: Plan de Pruebas del Sistema – Gestión de Proceso de Inventario

Caso de Prueba		GESTIÓN DE INVENTARIO	
Descripción		La prueba de Gestión de Inventario intenta probar la calidad del requerimiento de administrar el proceso de inventario y reporte de inventario con información de proveedores y productos.	
Precondiciones		No Aplica	
FUNCIONALIDAD	ENTRADA	RESULTADO ESPERADO	CUMPLE
Ingresar un nuevo proveedor	Se ingresa a la pantalla proveedor y se ingresa los datos del nuevo proveedor que son: idProveedor, nombre, direccion, telefono, correoElectronico y se presiona el botón Insertar	Se presenta un mensaje de Creado Exitosamente en caso de haberlo hecho o un mensaje de ERROR si no lo hace	SI
Modificar un proveedor	Se selecciona el proveedor a modificar y se reemplaza la información que se desea cambiar se presiona el botón Guardar	En caso de cambiar la información correctamente se presenta un mensaje de Cambio Exitoso caso contrario se presenta un mensaje de ERROR	SI
Consultar un proveedor	Se ingresa a la pantalla de proveedor y se escoge el proveedor a consultar	Se muestra la información del proveedor consultado	SI
Eliminar un proveedor	Se ingresa a la pantalla de proveedor se escoge el proveedor y se presiona el botón eliminar	En el caso de que el proveedor se haya eliminado correctamente se presenta un mensaje de Eliminado	SI

		Exitosamente caso contrario se presenta un mensaje de ERROR	
Ingresar un nuevo producto	Se ingresa a la pantalla producto y se ingresa los datos del nuevo producto que son: idProducto, descripcion, precio, stock, unidadMedida, cantidadMalEstado, cantidadTotal y se presiona el botón Insertar	Se presenta un mensaje de Creado Exitosamente en caso de haberlo hecho o un mensaje de ERROR si no lo hace	SI
Modificar un producto	Se selecciona el producto a modificar y se reemplaza la información que se desea cambiar se presiona el botón Guardar	En caso de cambiar la información correctamente se presenta un mensaje de Cambio Exitoso caso contrario se presenta un mensaje de ERROR	SI
Consultar un producto	Se ingresa a la pantalla de producto y se selecciona el producto a consultar	Se muestra la información del producto consultado	SI
Eliminar un producto	Se ingresa a la pantalla de producto se escoge el producto y se presiona el botón eliminar	En el caso de que el producto se haya eliminado correctamente se presenta un mensaje de Eliminado Exitosamente caso contrario se presenta un mensaje de ERROR	SI
Crear inventario	Se ingresa a la pantalla de reporte de ventas y se presiona el botón Crear Inventario	En el caso de que se cree correctamente el inventario se presenta un mensaje de Creado Correctamente en	SI

		otro caso se presente un mensaje de ERROR	
Eliminar Inventario	Se ingresa a la pantalla reporte de ventas, se selecciona el inventario a eliminar y se presiona el botón eliminar	En caso de ser eliminado correctamente se presenta un mensaje Eliminado Exitosamente en otro caso un mensaje de ERROR.	SI
Crear un reporte de Inventario	Se ingresa a la pantalla de reporte de ventas y se presiona el botón generar reporte de inventario.	En el caso de que se cree correctamente el reporte de inventario se presenta un mensaje de Creado Correctamente en otro caso se presente un mensaje de ERROR	SI

Fuente: Jorge Cunalata & David Morán

4.2.2 Informe de los resultados de las pruebas

Tabla 4: Resultado de Pruebas del Sistema – Gestión de Reserva de Mesas

Condiciones entradas	Entradas	Salidas Esperadas	Condiciones salida
Pantalla principal – archivo – opción reserva de mesas	Presionar la opción Reserva de mesas	Despliega ventana Reserva de Mesas	Se presentará la ventana reserva de mesas en otra ventana aparte de la principal

Se presentan las mesas disponibles con una imagen de una mesa de color verde	Las mesas disponibles se cargan dependiendo de la fecha y hora que se introduzca	Imagen de una mesa de color verde	El sistema presentará esta imagen dentro de la ventana de reserva de mesas
Se presentarán las mesas ocupadas con una imagen de una mesa de color rojo	Las mesas ocupadas se cargarán dependiendo de la fecha y la hora que se introduzca	Imagen de una mesa de color rojo	El sistema presentará esta imagen dentro de la ventana de reserva de mesas
Pantalla principal – archivo – opción reserva de mesas - se presentan los botones Crear y Consultar habilitados	Presionar la opción Reserva de mesas	Despliega ventana Reserva de Mesas	Se presentará la ventana reserva de mesas en otra ventana aparte de la principal
En la pantalla de reserva de mesas solo se habilitan los botones Eliminar y Modificar cuando previamente se haya realizado una consulta de una reserva	Presionar la opción Reserva de mesas y se realiza una consulta sobre una reserva con la fecha y cédula del cliente	Se habilitan los botones Eliminar y Modificar	Se habilitan los botones dentro de la pantalla de reserva de mesas
En la pantalla reserva de mesas al consultar una reserva	Se ingresa la fecha y la cédula del cliente y se presiona el botón Consultar	Se despliega la información de la reserva en las áreas de texto Fecha, Hora, Cantidad de personas, Cédula, Nombre, Teléfono	Se presenta la información de la reserva dentro de la pantalla de reserva de mesas

En la pantalla reserva de mesas al presionar el botón eliminar previo a realizar una consulta	Se presiona el botón eliminar previo al despliegue de la información tras realizar una consulta sobre una reserva y aparece una ventana de validación Está seguro de eliminar la reserva si la respuesta es sí se presiona Aceptar de lo contrario se presiona Cancelar	Aparece una ventana de validación para eliminar una reserva si se presiona si aparece una ventana con el mensaje Eliminado Exitosamente	Se presenta el mensaje de validación en una pantalla diferente a la de reserva de mesas al igual que se presenta un mensaje de éxito si acepta
---	---	--	--

Fuente: Jorge Cunalata & David Morán

Tabla 5: Resultado de Pruebas del Sistema – Gestión Proceso de Inventario

Condiciones entradas	Entradas	Salidas Esperadas	Condiciones salida
Pantalla principal – archivo – opción Inventario	Presionar la opción Inventario	Despliega ventana Inventario	Se presentará la ventana del proceso de Inventario que consta con tres pestañas que son Proveedores, Productos, Inventario
En la pestaña de proveedores se presentan el combo box para cargar proveedores, campos de texto de nombre, dirección, teléfono, e-mail y los botones Ingresar, Modificar, Eliminar y Cargar Proveedor	En el combo box se cargarán los proveedores existentes en caso de presionar el botón Cargar Proveedores, los campos de texto sirven para ingresar, modificar. Para eliminar un proveedor se seleccionará desde el combo box y se presionará el botón Eliminar.	Se presentan todos los campos y botones	El sistema siempre presentará esta pestaña al entrar a la interfaz de Inventario
En la pestaña de Productos se presentará el combo box para cargar productos, el combo para escoger el tipo de unidad de medida, los campos de taxto para descripción, precio, cantidad mal estado, cantidad total,	El combo box de productos cargará productos existentes cuando se presione el botón Cargar Producto, el combo box de unidad de medida tendrá varias opciones de unidades de medida para ingresar productos al igual que los campos de texto de	Se presentan todos los campos y botones	El sistema siempre presentará esta pestaña al entrar a la interfaz de Inventario

stock y los botones Ingresar, Cargar Producto, Modificar y Eliminar	descripción, precio, cantidad mal estado, cantidad total, stock. También se presentarán los botones Ingresar, Modificar y Eliminar		
En la pestaña de Inventario se presentan un combo box para cargar inventarios, un combo box para cargar reportes y tres botones Cargar Inventario, Cargar Reporte, Crear Inventario	El combo box de cargar inventario cargará inventarios existentes, el combo box de cargar reportes cargará reportes de inventarios existentes y el botón crear inventario creará inventario y reportes cuando se lo presione	Se presentan los dos combo box y los tres botones.	El sistema siempre presentará esta pestaña al entrar a la interfaz de Inventario

Fuente: Jorge Cunalata & David Morán

Tabla 6: Resultado de Pruebas del Sistema – Menú Principal

Condiciones entradas	Entradas	Salidas Esperadas	Condiciones salida
En la pantalla principal se presentará la barra de Menú con las opciones de Archivo y Ayuda	Al ingresar al sistema se presentará la pantalla del Menú Principal con una barra de herramientas que contendrá las opciones de Archivo y Ayuda	Despliega ventana del menú principal con una barra de herramientas	Se presenta la pantalla del Menú Principal
En la pantalla principal – opción archivo se presenta las opciones Inventario y Reserva de mesas	Se presentan las opciones de Inventario y Reserva de mesas dentro de la barra de herramientas opción Archivo	Se despliega las sub opciones de la barra de herramientas	Se presenta la pantalla del Menú Principal
En la pantalla principal – opción Ayuda se presenta las opciones Acerca de nosotros y Versión del producto	Se presentan las opciones de Acerca de nosotros y Versión del producto dentro de la barra de herramientas opción Ayuda	Se despliega las sub opciones de la barra de herramientas	Se presenta la pantalla del Menú Principal

Fuente: Jorge Cunalata & David Morán

5. FASE DE TRANSICIÓN

5.1 Despliegue

5.1.1 Software empaquetado

Para el empaquetado y posterior instalación del software realizado se utilizó la herramienta Inno Setup que es un instalador gratuito para programas ejecutables en Windows.

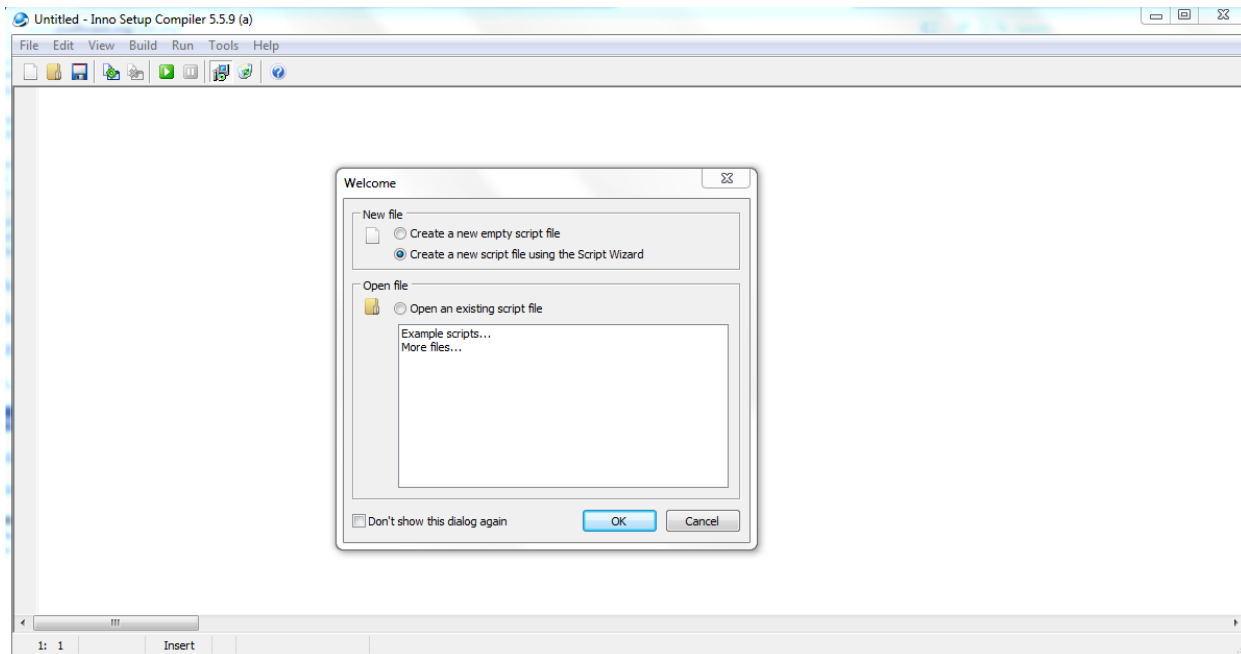
Las ventajas que tiene la herramienta Inno Setup es que soporta todo sistema operativo Windows desde el año 2000 aparte que realiza ejecutables para 32 y 64 bits tiene una interfaz muy amigable con el usuario al momento de realizar la instalación.

Mediante el script generado por la herramienta Inno Setup se genera el archivo para la ejecución de un .exe para proceder con la instalación del programa.

Los pasos que se siguieron para el empaquetado del software fueron los siguientes:

- Se ejecuta la herramienta Inno Setup previa a su instalación.

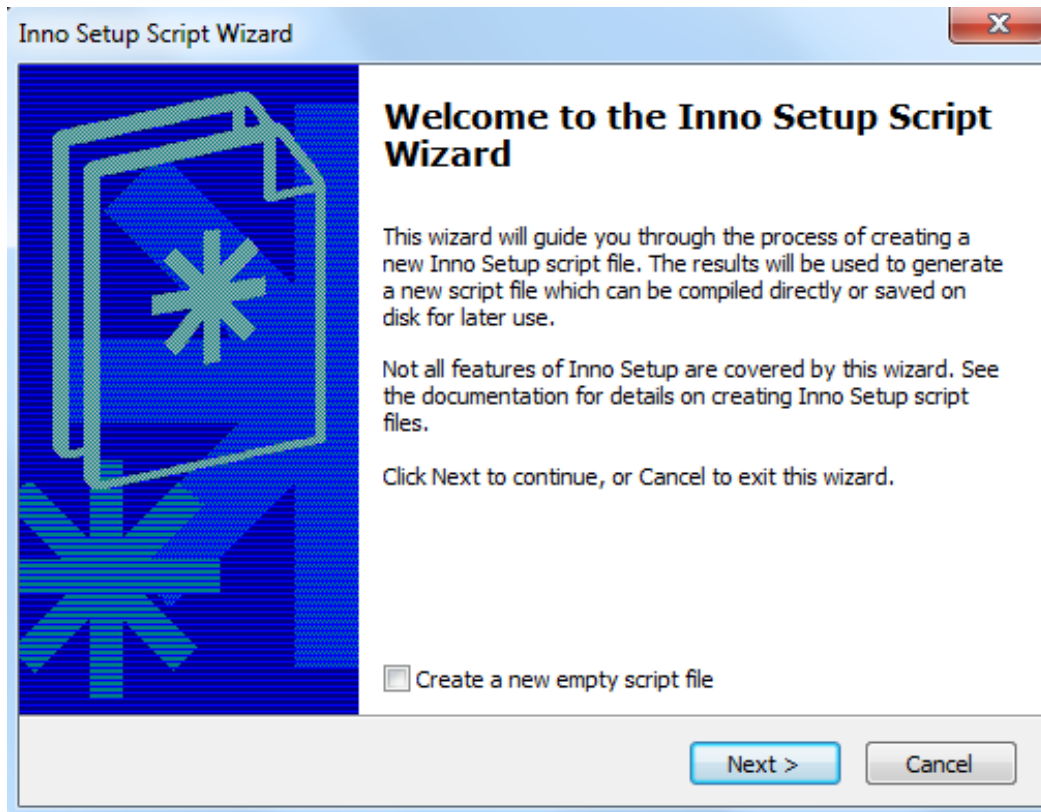
Ilustración 50: Empaquetado del Programa y Generación del Ejecutable



Fuente: Jorge Cunalata & David Morán

- Se selecciona la opción crear un nuevo script usando el asistente de generación de scripts.

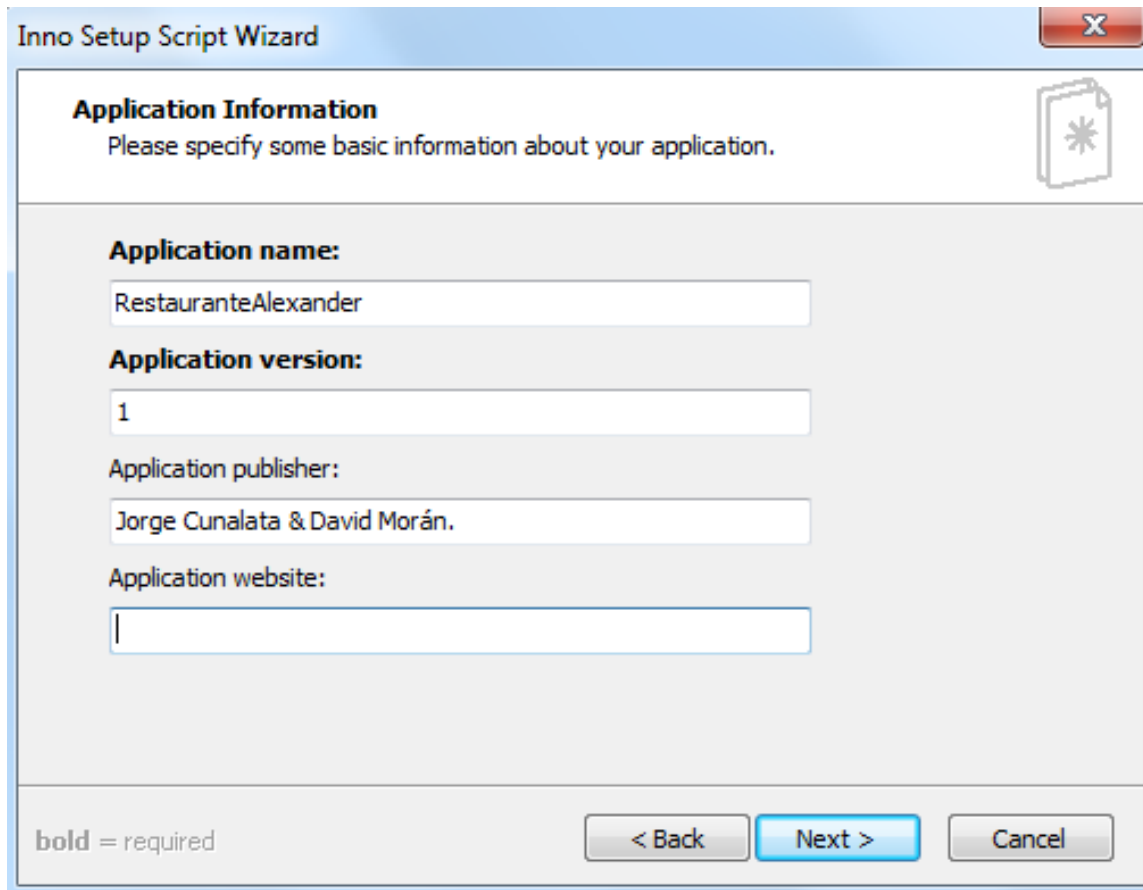
Ilustración 51: Empaquetado del Programa y Generación del Ejecutable



Fuente: Jorge Cunalata & David Morán

- Se escoge las opciones sobre la información de la aplicación como nombre de la aplicación, versión, autores de la aplicación, el .Java a ser empaquetado y los componentes extras como el JRE de Java, Postgres y su JDBC para la comunicación con la base de datos.

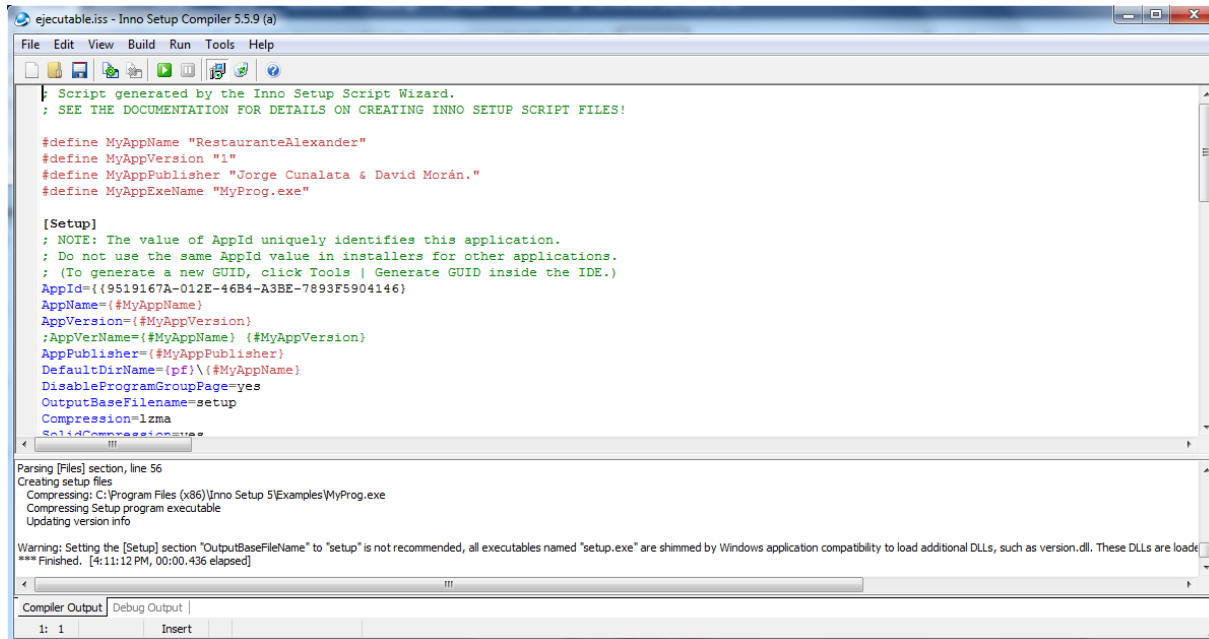
Ilustración 52: Empaquetado del Programa y Generación del Ejecutable



The image shows a screenshot of the 'Inno Setup Script Wizard' dialog box, specifically the 'Application Information' step. The window title is 'Inno Setup Script Wizard'. The main heading is 'Application Information' with a sub-instruction: 'Please specify some basic information about your application.' Below this, there are four input fields: 'Application name:' with the text 'RestauranteAlexander', 'Application version:' with the text '1', 'Application publisher:' with the text 'Jorge Cunalata & David Morán.', and 'Application website:' which is currently empty. At the bottom left, it says 'bold = required'. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted in blue.

Fuente: Jorge Cunalata & David Morán

Ilustración 53: Empaquetado del Programa y Generación del Ejecutable



Fuente: Jorge Cunalata & David Morán

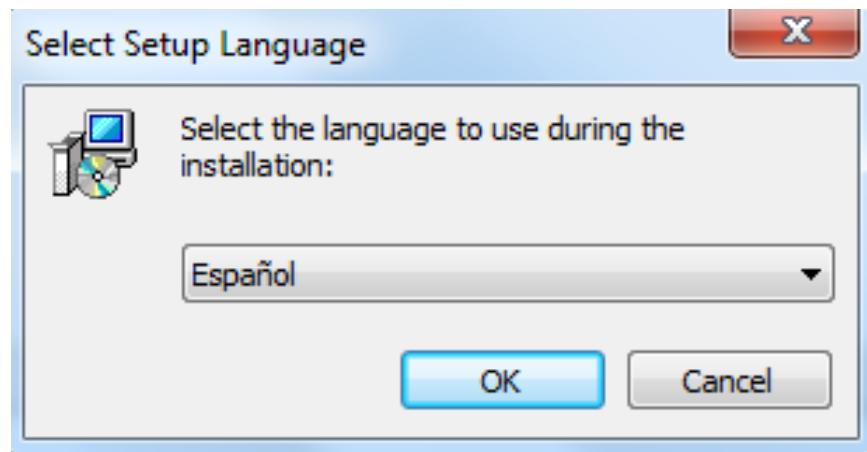
5.1.2 Instalación del software

Para la instalación del programa del Restaurante y Servicio de Cáterin Alexander se debe tener en cuenta que dentro del empaquetado del software debe estar el JRE requerido así como el JDBC para la conexión con la base de datos que en este caso fue Postgres. Se puede también adjuntar el instalador de Postgres para la generación de la base de datos.

Para su instalación solo se tienen que seguir los siguientes pasos

- Una vez creado el ejecutable (.exe) lo probamos y su resultado nos da lo siguiente
- Se abre un asistente de instalación de software muy fácil de seguir en el idioma que se desee.

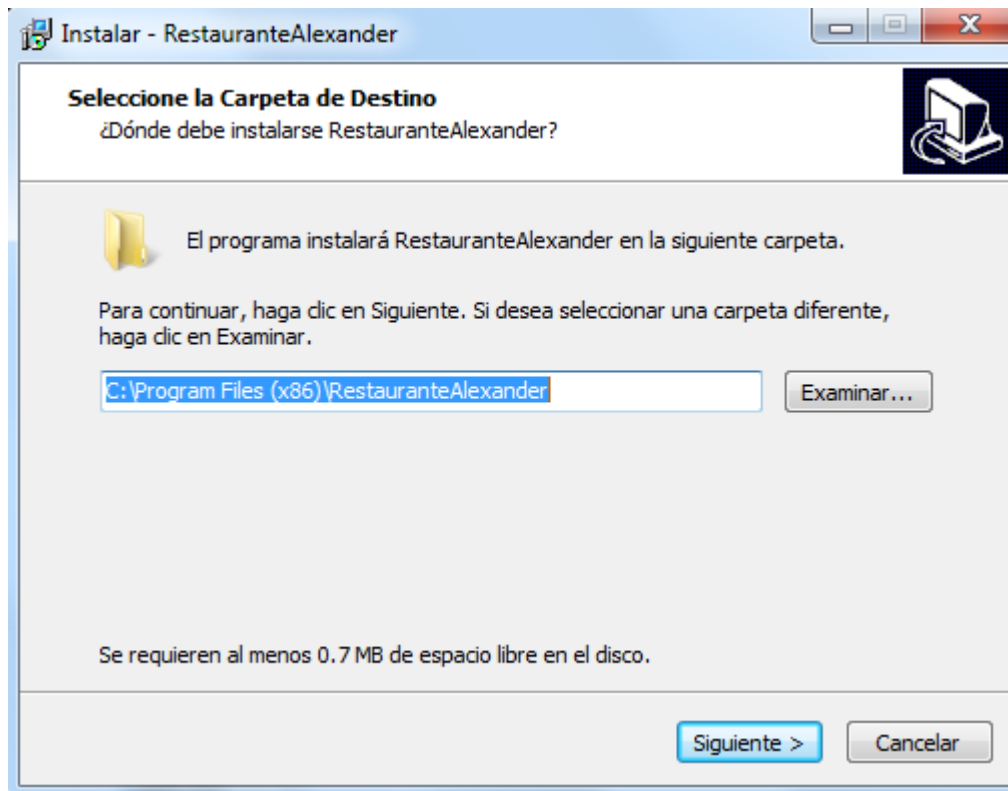
Ilustración 54: Instalación del software RestauranteAlexander



Fuente: Jorge Cunalata & David Morán

- Seleccionamos la ubicación donde queremos que la aplicación se instale.

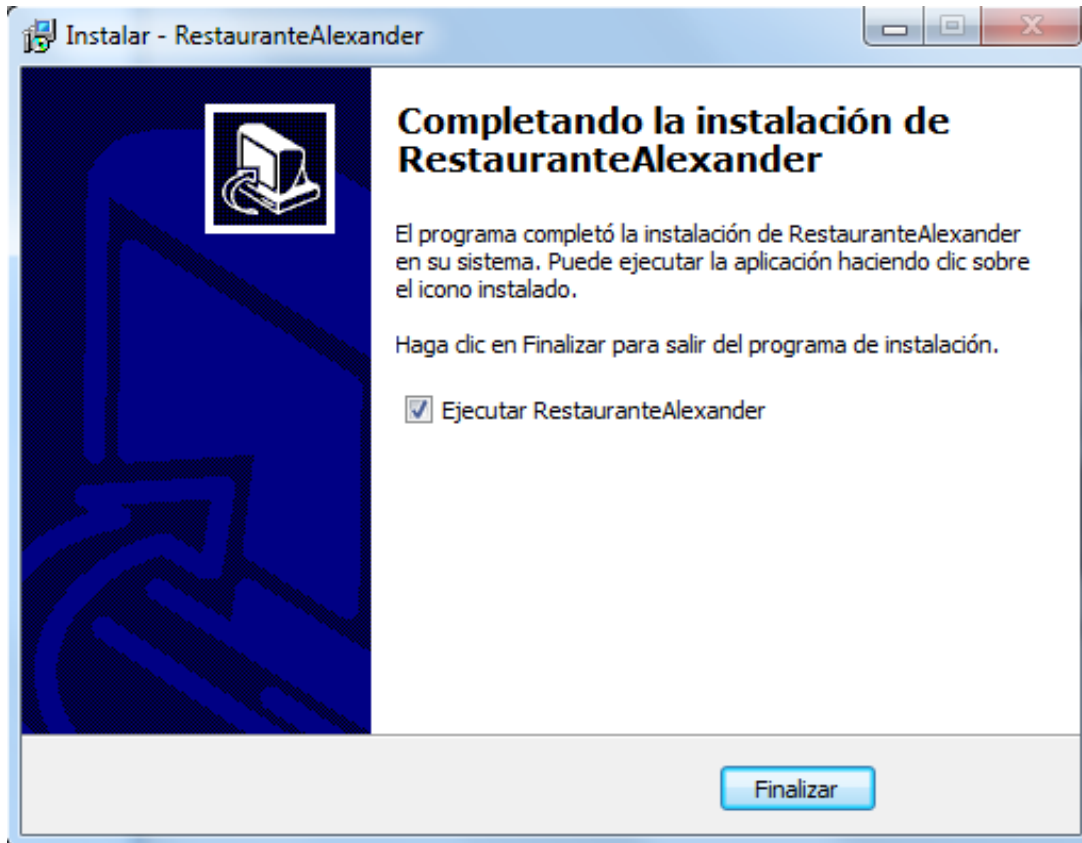
Ilustración 55: Instalación del software RestauranteAlexander



Fuente: Jorge Cunalata & David Morán

- Termina la instalación y le damos a la opción finalizar.

Ilustración 56: Instalación del software RestauranteAlexander



Fuente: Jorge Cunalata & David Morán

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- Se utilizó la metodología de desarrollo RUP ya que esta es un poco más flexible a las necesidades del cliente.
- Se aseguró la calidad tanto de los entregables como del producto final mediante de la utilización de la metodología RUP y UML para el levantamiento de los principales procesos.
- El desarrollo del programa sobre una arquitectura de tres capas dio como resultado un producto final fácil de integrar y con una mantenibilidad futura no compleja.
- Se utilizó el IDE NetBeans de Java ya que es un entorno de desarrollo gratuito que soporta varias tecnologías y muy intuitiva de usar en cuanto a su parte gráfica para la creación de las pantallas.
- La comunicación constante con el cliente facilitó el entendimiento del flujo del negocio y esto nos ayudó para el levantamiento de los principales procesos y el desarrollo del software.
- El haber realizado un análisis detallado muy minucioso fue la base correcta para comenzar con la construcción del software.
- La herramienta de empaquetado de software Inno Setup permitió generar el archivo .exe de una forma muy intuitiva y con todo lo necesario para que el sistema funcione.

6.2 Recomendaciones

- Se recomienda mantener una constante comunicación con el cliente en todos los ciclos del desarrollo del producto.
- Se recomienda documentar y comentar el código generado para un posterior alcance del mismo o facilitar la mantenibilidad del mismo.
- Se recomienda crear Backup de todos los entregables que se vayan generando.
- Se recomienda investigar en el análisis detallado sobre la compatibilidad de las herramientas a utilizar en el desarrollo y sus versiones recomendadas.
- Se recomienda que en el empaquetado del producto final sí incluyan todas las herramientas para que el programa funcione correctamente.
- Se recomienda crear un manual de usuario fácil de entender, sin ambigüedades y muy interactivo para que cualquier persona lo pueda seguir.
- Se recomienda dar un curso pos entrega del proyecto sobre su funcionamiento.
- Realizar las pruebas funcionales y técnicas lo más minucioso que se pueda para detectar defectos.

Bibliografía

- International Technical Support Organization. (2007) “*The IBM Rational Unified Process for System Z*” [Libro en línea]. [US], URL:<http://www.redbooks.ibm.com/redbooks/pdfs/sg247362.pdf>.
- Sierra, Kathy; Bates, Bert.(2008) “*SCJP Sun® Certified Programmer for Java™ 6 Study Guide*” [Libro en línea]. [US], URL:<http://firozstar.tripod.com/darksiderg.pdf>.
- Ahmed, Ibrar; Fayyaz, Asif; Shahzad, Amjad.(2015)” *PostgreSQL Developer's Guide*” [Libro en línea]. [US], URL:https://www.packtpub.com/big-data-and-business-intelligence/postgresql-developers-guide/?utm_source=PoD&utm_medium=referral&utm_campaign=1783989025.
- Stephen B. Shiring, SR; Jardine, Williams; Richard J. Mills, Jr. (2001) “ *Introduction to Catering: Ingredients for success* ” [Libro en línea]. [US], URL:
<https://books.google.com.ec/books?id=k4QsU130gjsC&pg=PA77&dq=catering+services&hl=es&sa=X&ved=0CCEQ6AEwAWoVChMIrv6VyLWLyAIVTBYeCh01WAUs#v=onepage&q=catering%20services&f=false>

Glosario

- **Software:** Es un conjunto de instrucciones que los procesadores emplean para manipular datos.
- **IBM:** International Business Machines empresa dedicada al desarrollo de software y hardware.
- **Metodología:** Etapa dentro de un proyecto en la que se guía mediante métodos y teoría para la realización de ciertas tareas.
- **Artefacto:** Es un resultante tangible dentro del proceso de desarrollo de software.
- **UML:** Lenguaje Unificado de Modelado es un estándar para crear esquemas, diagramas y documentación en el desarrollo de software.
- **Casos de uso:** Diagrama en el que se indica las transacciones que realiza un actor dentro de un Sistema.
- **Arquitectura:** Es la relación entre las partes que constituyen un Sistema.
- **Producto beta:** Es un producto lanzado a los usuarios para pruebas públicas.
- **Ingeniería de software:** Es una disciplina que se ocupa del desarrollo, la operación y el mantenimiento de software.
- **Lenguaje de programación:** Son todos los lenguajes formales diseñados para realizar procesos que puedan ser interpretados por los ordenadores.
- **Programas ejecutables:** Son programas los cuales pueden ser llevados a cabo por una computadora.
- **Sun Microsystems:** Era una compañía Norte Americana dedicada a la venta de computadores y componentes de computadores.
- **Oracle Corporation:** Es una compañía de software que desarrolla base de datos.