



**UNIVERSIDAD DE ARTES, CIENCIAS Y COMUNICACIÓN  
Facultad de Administración  
Carrera de Ingeniería Informática Multimedia**

**Proyecto para optar al Grado Académico de Licenciado/a en  
Ingeniería Informática Multimedia y al Título Profesional de Ingeniero  
Informático Multimedia**

**Profesor Guía:**

**Ricardo Hernán Valenzuela Castro**

**Estudiantes:**

**Sara Coñuen R. Roxana Pérez G. Miguel Carbone W. Gerardo Gallardo R.**

**Santiago de Chile, 17 diciembre 2020**

# Índice

1. Capítulo I: Planteamiento de la Problemática .....	8
1.1. Resumen Ejecutivo .....	8
1.2. Antecedentes del Proyecto .....	8
1.3. Definición del Problema .....	10
2. Capítulo II: Referentes o marco teórico.....	12
2.1. Bases Teóricas: Diseño lógico especificación de requerimientos .....	12
3. Capítulo III: Referentes o marco metodológico .....	19
3.1. Metodología utilizada .....	19
3.1.1. Diagramas del Proyecto - Metodología UML .....	20
3.1.2. Diagramas de Caso de uso .....	21
3.1.3. Diagramas de Secuencias.....	23
3.2. Beneficios de la metodología Waterfall.....	25
3.3. Metodología desarrollo de software .....	25
3.4. Diccionario de datos.....	33
4. Capítulo IV: Planificación del proyecto.....	35
4.1. Cronograma de actividades .....	35
4.2. Análisis FODA ¿cuáles son las fortalezas, debilidades, oportunidades y amenazas de su proyecto?.....	36
4.2.2. Debilidades .....	36
4.2.3. Oportunidades.....	37
4.2.4. Amenazas .....	37
4.3. Estudio de Factibilidad .....	37
4.3.1. Factibilidad Técnica.....	38
4.3.2. Factibilidad Económica.....	40
4.3.3. Factibilidad Legal.....	42
4.3.4. Factibilidad de Recursos .....	43
4.3.5. Factibilidad de Mercado.....	44
4.3.6. Factibilidad Operacional .....	46
4.3.7. Factibilidad de Tiempo.....	47
4.3.8. Check List de aseguramiento de calidad .....	48
4.3.9. Análisis de Riesgos .....	49
4.4. Cuantificación del Proyecto en esfuerzos y costos.....	50
4.5. Justificación viable del Proyecto .....	51
4.5.1. Justificación de las herramientas de desarrollo utilizadas.....	52
4.6. Evaluación de alternativas en el mercado.....	54
4.6.1. Posible Competencias en el mercado actual .....	54
4.6.2. Comparativo entre las funciones principales de las 3 aplicaciones de Gestión de Restaurant .....	57

4.6.3. Uso de Tecnologías .....	62
4.6.4. Software Propio.....	62
4.6.5. Desarrollo Externo.....	63
4.6.6. Operación de Sistema Externo .....	64
4.6.7. Explotación de Sistemas Externos BPO .....	64
4.6.8. Desarrollo de software .....	65
4.7. ¿Qué aspectos debe contemplar una prueba bien definida? .....	65
4.8. ¿Que son las pruebas automatizadas?.....	66
4.8.1. Ventajas Pruebas Automatizadas.....	67
4.8.2. Desventajas Pruebas Automatizadas .....	68
4.9. Pruebas Unitarias e Integración .....	68
4.9.1. Pruebas Unitarias - Alcance .....	69
4.9.2. Pruebas de integración – Alcance .....	69
4.10. Pruebas automatizadas con JUnit .....	70
4.11. Pruebas de Sistema.....	83
4.11.1. Alcance.....	83
4.12. Pruebas de Caja Negra .....	86
4.14. Mejora Continua para el proyecto SoftCook Management .....	87
4.14.1. Mejoras contempladas .....	88
4.14.2. Impacto post implementación.....	89
5. Logros Obtenidos.....	90
6. Referencias.....	91
7. Conclusiones y Reflexiones .....	92

## Índice de Figuras

Figura 1: Información extraída Censo 2017 .....	10
Figura 2: Diseño lógico.....	12
Figura 3: Definición de requerimientos.....	18
Figura 4: APP_Cocina.....	18
Figura 5: App_Stock.....	18
Figura 6: Diagrama UML App_Cocina.....	20
Figura 7: Diagrama UML App_Stock.....	21
Figura 8: Diagrama Caso de Uso – Caso 1.....	21
Figura 9: Diagrama Caso de Uso – Caso 2.....	22
Figura 10: Diagrama Caso de Uso – Caso 3.....	23
Figura 11: Diagrama Secuencia 1.....	24
Figura 12: Diagrama Secuencia 2.....	24
Figura 13: Diagrama Secuencia 3.....	25
Figura 14: Conexión Base de Datos.....	26
Figura 15: Librería MySQL.....	26
Figura 16: Ejecución XAMPP.....	27
Figura 17: Ingreso PhpMyAdmin.....	27
Figura 18: Ejecución APP_Cocina - Selección.....	28
Figura 19: Ejecución APP_Cocina - Ingredientes.....	28
Figura 20: Ejecución APP_Cocina Mensaje.....	29
Figura 21: Ejecución App_Stock - Existencias.....	30
Figura 22: Ejecución App_Stock - Agregar.....	30
Figura 23: Ejecución App_Stock - Editar.....	31
Figura 24 : Ejecución App_Stock - Eliminar.....	31
Figura 25: Pruebas Integración.....	32
Figura 26: Pruebas Integración.....	32
Figura 27: Pruebas Integración - Mensaje.....	33
Figura 28: Diccionario de Datos – Cliente/Detalle Plato.....	33
Figura 29: Diccionario de Datos – Ingredientes.....	33
Figura 30: Diccionario de Datos – Plato.....	34
Figura 31: Diccionario de Datos – Índices.....	34
Figura 32 : Carta Gantt.....	35
Figura 33: Estudio de factibilidad – Alcance.....	38
Figura 34: Esquema Base de Datos.....	40
Figura 35: Uso de Tecnologías.....	62
Figura 36: Prueba automatizada.....	70
Figura 37: Realización de Prueba.....	71
Figura 38: Librerías.....	71
Figura 39: Test Insertar Ingrediente.....	72
Figura 40: Test Actualización Ingredientes.....	72
Figura 41: Test Eliminación Ingredientes.....	72
Figura 42: Realización Test.....	73
Figura 43: Revisión Pruebas.....	73
Figura 44: Prueba de Inserción.....	74

Figura 45: Revisión de Base Datos .....	74
Figura 46: Realización Test Actualización.....	75
Figura 47: Revisión Base de Datos .....	75
Figura 48: Realización Test Eliminación .....	76
Figura 49: Revisión Test Eliminado.....	76
Figura 50: Pruebas Unitarias App_Cocina y App_Stock .....	77
Figura 51: Opciones para realizar test .....	77
Figura 52: Llamado Librerías.....	78
Figura 53: Métodos – Test Insertar .....	78
Figura 54: Métodos – Test Actualizar .....	78
Figura 55: Test Eliminación .....	79
Figura 56: Realización Test.....	79
Figura 57: Revisión Base Datos .....	80
Figura 58: Realización Inserción .....	80
Figura 59: Revisión Base Datos en phpMyAdmin .....	81
Figura 60: Test Actualización Realizado .....	81
Figura 61: Revisión en Base de Datos .....	82
Figura 62: Realización Test Eliminación .....	82
Figura 63: Revisión Test Eliminado en phpMyAdmin .....	83
Figura 64: Muestra de platos a preparar .....	84
Figura 65: Mensaje de plato preparado.....	84
Figura 66: Revisión de existencias.....	85
Figura 67: Revisión Implementación .....	86
Figura 68: Ejemplo de mejora continua contemplada 1 .....	89
Figura 69: Ejemplo de mejora continua contemplada 2 .....	89

## Índice de Tablas

Tabla 1: APP_Cocina .....	14
Tabla 2: App_Stock .....	15
Tabla 3: Entradas APP_Cocina.....	16
Tabla 4: Entradas App_Stock.....	16
Tabla 5: Salida APP_Cocina .....	17
Tabla 6: Salida App_Stock .....	17
Tabla 7: Beneficios metodología .....	25
Tabla 8: Factibilidad Económica – Costo de Personal .....	41
Tabla 9: Factibilidad Económica – Costo de Mantenición .....	41
Tabla 10: Factibilidad Económica – Costo totales.....	42
Tabla 11: Factibilidad de Recursos – Desarrollo del Proyecto .....	44
Tabla 12: Factibilidad Operacional .....	47
Tabla 13: Factibilidad de Tiempo .....	48
Tabla 14: Check List.....	49
Tabla 15: Valores del Riesgo .....	49
Tabla 16: Análisis del Riesgo .....	50
Tabla 17: Cuadro comparativo .....	53
Tabla 18: Cuadro comparativo de funcionalidades .....	58
Tabla 19: Costos del Plan Standard - SmartRestoBar .....	59
Tabla 20: Costos del Plan Premium - SmartRestoBar .....	59
Tabla 21: Costos del Plan Smart - SmartRestoBar .....	60
Tabla 22 : Costos Plataforma Resto.....	61
Tabla 24: Aspectos de pruebas bien definidas.....	66
Tabla 25: Prueba de Caja Negra App_Cocina .....	87
Tabla 26: Prueba de Caja Negra App_Stock.....	87

## Introducción

Debido a la creciente propagación de Internet, hoy en día es recomendable usar este medio para acercarse a futuros clientes, se puede proyectar un sinfín de ideas para ayudar a mejorar la ejecución diaria de las labores domésticas del Restaurant Tayu, el mayor problema generalizado en cadenas de restaurant es la falta de orden y control de ingredientes, de mantener una bodega 100% equipada y no caer en la problemática de no tener stock de algún ingrediente para posterior no poder realizar la preparación de un plato.

Restaurant Tayu, no está ajeno a la realidad antes descrita, y para abordar los problemas del manejo de stock y mermas es que se desarrollaron dos aplicaciones en Java, donde ambas interactúan con una base de datos desarrollada en SQL, las cuales se implementarán en este restaurant, que requiere administrar de mejor manera su negocio, tanto de cocina como de stock, este la denominamos “**SoftCook Management**”

La primera aplicación creada es **Cocina** y permitirá administrar los diferentes ingredientes de los platos que se preparan en el restaurant.

La segunda aplicación es **Stock** y ésta le permitirá conocer al restaurant el nivel de stock de los productos para así poder gestionar la adquisición de los ingredientes que están faltando.

# **1. Capítulo I: Planteamiento de la Problemática**

## **1.1. Resumen Ejecutivo**

Este proyecto propone crear una solución informática llamada SoftCook Management, para el Restaurant Tayu, ubicado en la Región de la Araucanía específicamente en Villarrica, la cual nos permitirá obtener una aplicación que nos ayudará a mejorar el control del stock de las bodegas que posee.

Para poder realizar este proyecto lo primordial será analizar y aplicar los conocimientos adquiridos en todo nuestro proceso académico con la Universidad Uniacc, y para esto se usarán diferentes metodologías y poder llevar a cabo esta nueva propuesta de solución informática.

## **1.2. Antecedentes del Proyecto**

### **Visión:**

Nuestra visión es entregar herramientas informáticas para facilitar la gestión y eficiencia del Restaurant Tayu, para lograrlo se abordará la problemática del NO control de ingredientes en la bodega actual que posee Restaurant, en base a esto el sistema creado debe ser capaz de solucionar esta problemática y juntamente proporcionar orden al stock de cada producto e ingrediente.

### **Origen del proyecto:**

Dada la gran necesidad del Restaurant Tayu, por no poseer el control correcto de manejo de ingredientes de la cocina que administran, es que nace la necesidad de creación de este proyecto denominado SoftCook Management, el propósito en general es evitar mermas y un control exacto de lo que mantienen en bodega para la correcta preparación de platos, y así poder siempre mantener la cantidad de público que se necesita para existir.

El equipo de proyectos decide la creación de dos aplicaciones que se relacionan entre sí, que tenga la facultad de estar integradas a través de una base de datos normalizada en SQL.



Los resultados esperados con este proyecto es dar solución a la problemática que presenta el restaurant, en donde requiere de dos sistemas informáticos para su funcionamiento, donde el primer sistema será el encargado de administrar los diferentes ingredientes de los platos que se preparan en el restaurante, el segundo sistema permitirá conocer el nivel de stock de los productos del restaurante, para así poder gestionar la adquisición de los ingredientes que están faltando.

Además, este proyecto debe contar con una base de datos que interactúe con ambos sistemas, la cual debe cubrir el requerimiento del negocio propuesto.

### **Antecedentes del Restaurant Tayu:**

El Restaurant Tayu de origen Mapuche, ubicado en Villarrica, ubicada en la provincia de Cautín parte de la Región de la Araucanía.

Actualmente Villarrica cuenta con alrededor de 55.478 habitantes, fundada en el año 1952 por Gerónimo de Alderete, esta localidad ha sufrido de muchos obstáculos en su historia que importunaron su estabilidad político-administrativa en Chile, y a pesar de esto, ha tenido un crecimiento constante que la ha dejado como una de las comunas más importantes de la zona, tanto por su oferta de servicios, cantidad de población y su incipiente turismo y falta de ofertas gastronómicas.

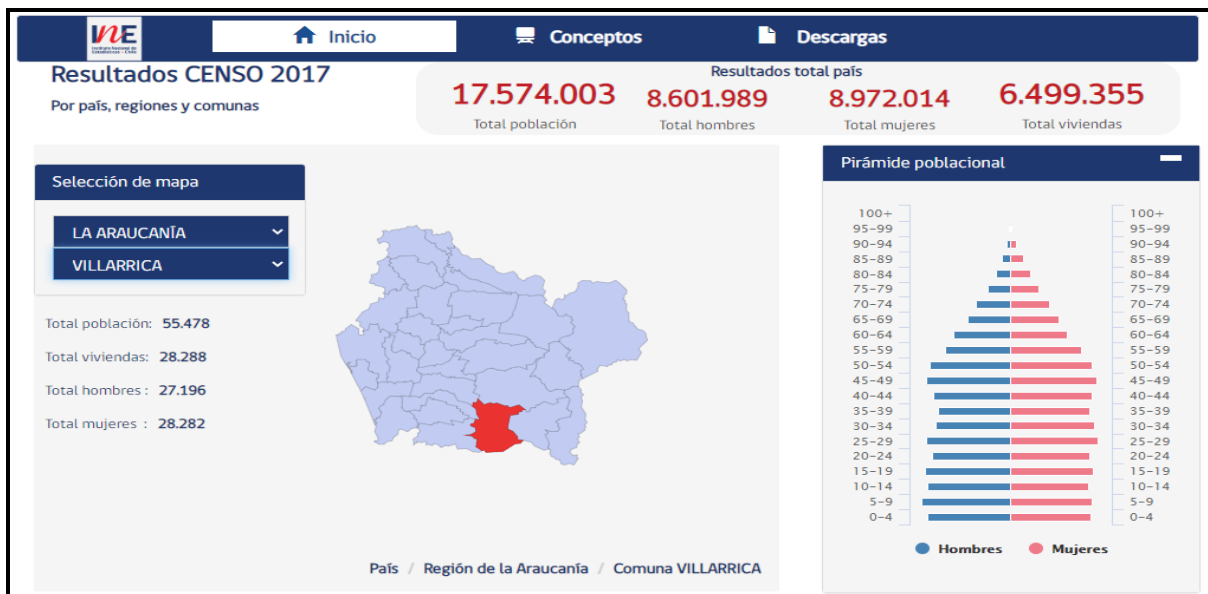


Figura 1: Información extraída Censo 2017

### 1.3. Definición del Problema

El restaurant Tayu, no cuenta con un control de stock de ingredientes que mantienen en la cocina, dada esta necesidad es la propuesta presentada en este informe.

Hoy en día cualquier negocio incluyendo restaurantes compiten por ser el mejor, ofrecer un servicio rápido, de calidad, fiable y con el menor costo posible, el personal del restaurant sólo ansia tener un trabajo administrativo, rápido y ordenado, eso quiere decir que nos debemos enfocar en la correcta gestión del inventario, ya que juega un papel fundamental para lograrlo, una rotura de stocks o cualquier otra incidencia puede significar la pérdida de una venta o la preparación de un plato de forma incorrecta. Se debe pensar en un cliente que llegó al restaurant y solicita de un plato que se encuentra en la carta, y tras haber realizado su elección, el mesero o encargado se da cuenta que el plato hoy no se encuentra disponible, ya sea por falta de stock de ingredientes terminará en que el cocinero encargado no podrá prepararlo, esta acción es la que debemos evitar registrando todo en nuestro sistema propuesto.

Todos estos problemas se deben a una mala gestión de inventario y puede estropear notablemente la nueva imagen corporativa de cara a un posible cliente, para esto el control de stocks es un elemento esencial a la hora de gestionar un negocio ya sea de cualquier rubro.

El control del inventario forma parte de una logística de la empresa o restaurant y consiste en organizar, planificar y controlar un conjunto de “mercancías” que deben siempre estar disponible para la preparación de los platos.

El objetivo es, por lo tanto, garantizar que siempre un cliente del restaurant solicite un plato de la carta y se pueda proporcionar en el tiempo y forma esperada.

## 2. Capítulo II: Referentes o marco teórico

### 2.1. Bases Teóricas: Diseño lógico especificación de requerimientos

A continuación, se dará a conocer una descripción del comportamiento del sistema desarrollado donde se incluirá descripción del proceso, Validaciones de datos y procesos, requerimientos funcionales y no funcionales, entradas y salidas y diagramas.

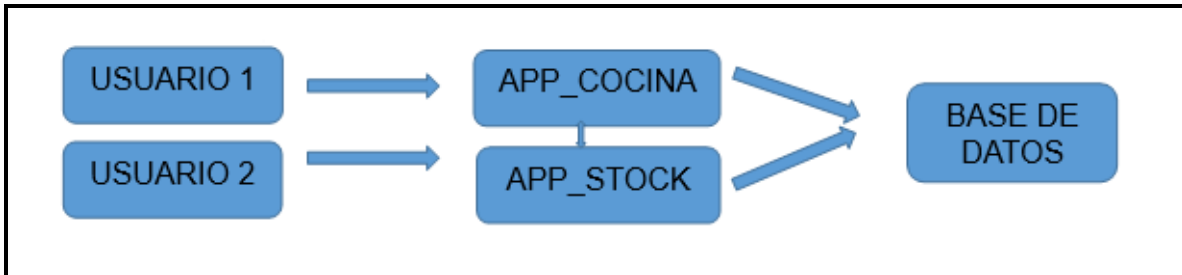


Figura 2: Diseño lógico

#### Descripción del proceso – Alcances:

Dentro de los alcances del sistema SoftCook Management, se encuentra la eficiencia del programa, permitiendo los resultados esperados del proyecto.

Según las necesidades recolectadas, se desarrollan las siguientes funciones del sistema:

- Gestionar la información de los ingredientes: buscar, editar, modificar y eliminar.
- Gestionar el nivel de stock de ingredientes: buscar, editar, modificar y eliminar.
- Gestionar el tipo de pago (en tipo de moneda): Buscar.

#### Informes:

En esta sección se muestra los informes que se generan al momento de editar o modificar algún dato, tanto de los ingredientes como del stock.

- Generar informe del ingreso o eliminación de los ingredientes
- Generar informe del stock actualizado de los ingredientes
- Generar informe de la utilización correcta del pago

## **Validación de datos y procesos:**

A continuación, se mostrarán los elementos esenciales para la validación de datos y procesos del proyecto del sistema.

- **Interfaces de usuario:** Las interfaces consideraran una vista principal donde el usuario podrá ver el menú general del sistema SoftCook Management, en la segunda vista podrá encontrar los platos que se preparan en el restaurant de los cuales, al presionarlos se desplegará una lista con cada ingrediente, al momento de presionar el ingrediente, éste, automáticamente se descontará de la base de datos. El usuario tiene la opción de volver al menú principal cada vez que lo requiera.
- **Interfaces de stock:** El usuario, en la primera vista podrá encontrar los tipos de platos que se preparan junto a los ingredientes que usa, en la segunda vista podrá encontrar la cantidad que dispone cada ingrediente, con la posibilidad de cambiar, agregar o quitar del stock.
- **Errores:** Los errores que puede haber dentro del procesos, es que el usuario puede agregar algún ingrediente que ya se encuentre en el stock. Para estos problemas el sistema arrojará un informe advirtiendo la existencia de dichos productos. Ante errores del sistema, es necesario la solicitud del ingeniero para dar solución a los errores de sistema.
- **Ambiente de uso:** es necesario que el ambiente de uso permita ejecutar programas basados en JAVA, en cuanto al hardware Procesador 2.0 GHz, 4 Gb de RAM, 500 Gb disco duro y S.O Windows/MAC OS.
- **Requisitos de rendimiento:** Es necesario que el sistema responda de manera rápida ante la solicitud de los ingredientes de cada plato y así también ante la respuesta del descuento de la base de datos desde la vista o APP1 a la vista o APP2 (stock).

Para el aseguramiento de la validación de datos y procesos, concluyendo la realización del sistema el fabricante deberá tener los siguientes documentos.

- La definición de los requisitos del usuario
- El protocolo de validación utilizado
- Los criterios de aceptación
- Las pruebas y los resultados
- Un resumen de validación

**Requerimientos funcionales:**

Los requerimientos funcionales detallan los requisitos fundamentales del cliente, para que el sistema SoftCook Management sea funcional y de calidad.

**TABLA1 – APP\_Cocina:**

NOMBRE	DESCRIPCIÓN
Búsqueda de información de un ingrediente determinado.	La aplicación debe permitir buscar la cantidad en la base de datos dado el nombre de un ingrediente y mostrarla en pantalla.
Ingreso a la base de dato del stock desde la App_Cocina.	Descuento del stock del ingrediente solicitado y mostrarlo en pantalla.

*Tabla 1: APP\_Cocina*

**TABLA2 – APP\_Stock**

NOMBRE	DESCRIPCIÓN
--------	-------------

Búsqueda de información de un ingrediente determinado.	La aplicación debe permitir buscar la cantidad e información del ingrediente solicitado, permitiendo la modificación o eliminación de él en la base de datos.
Modificación de Ingrediente	La aplicación debe permitir la modificación de la información o cantidad del ingrediente, mostrando por pantalla la realización exitosa o rechazo de la acción.
Agregar Ingrediente	La aplicación debe permitir la inserción de nuevos ingredientes y cantidad de estos al sistema, mostrando por pantalla la realización exitosa o rechazo de la acción.
Eliminar Ingrediente	La aplicación debe permitir la eliminación de ingredientes y cantidad de estos del sistema, mostrando por pantalla la realización exitosa o rechazo de la acción.

Tabla 2: App\_Stock

### Requerimientos no funcionales:

Ingreso del usuario al sistema es libre sin restricción por contraseña.

Al momento de presionar el ingrediente desde la App\_Cocina el descuento de la cantidad del stock es inmediato sin opción a modificación por parte del usuario de Cocina.

El sistema no cuenta con un informe diario de la cantidad de ingredientes entrantes y salientes del stock.

### ENTRADAS APP\_COCINA:

En la Tabla se detalla las interfaces de usuario y los datos a ingresar en cada caso.

NOMBRE		DETALLE DE DATOS
RE_1	Buscar información de ingrediente	Nombre y código del ingrediente
RE_2	Descontar stock de ingrediente	Cantidad y nombre del ingrediente

Tabla 3: Entradas APP\_Cocina

#### ENTRADAS APP\_STOCK:

NOMBRE		DETALLE DE DATOS
RE_1	Buscar información de ingrediente	Nombre y código del ingrediente
RE_2	Descontar stock de ingrediente	Cantidad y nombre del ingrediente
RE_3	Ingresar nuevo ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>
RE_4	Modificar Ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>
RE_5	Eliminar Ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>

Tabla 4: Entradas App\_Stock

#### SALIDA APP\_COCINA

NOMBRE	DETALLE DE DATOS	SALIDA
--------	------------------	--------



<b>RS_1</b>	Listar ingredientes	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_2</b>	Informe de stock de ingredientes	<ul style="list-style-type: none"> <li>• Código</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_3</b>	Informe descuento de Ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> </ul>	PANTALLA

Tabla 5: Salida APP\_Cocina

### SALIDA APP\_STOCK

	<b>NOMBRE</b>	<b>DETALLE DE DATOS</b>	<b>SALIDA</b>
<b>RS_1</b>	Listar ingredientes	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_2</b>	Informe de stock de ingredientes	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_3</b>	Informe ingreso ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_4</b>	Informe de modificación	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA
<b>RS_5</b>	Informe de Eliminación ingrediente	<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Cantidad</li> </ul>	PANTALLA

Tabla 6: Salida App\_Stock

### Diagrama 1: Definición de requerimientos

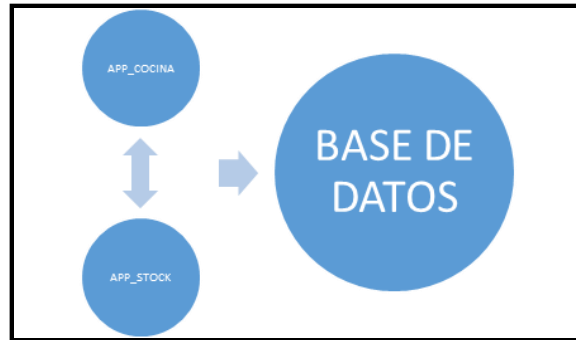


Figura 3: Definición de requerimientos

### APP\_COCINA

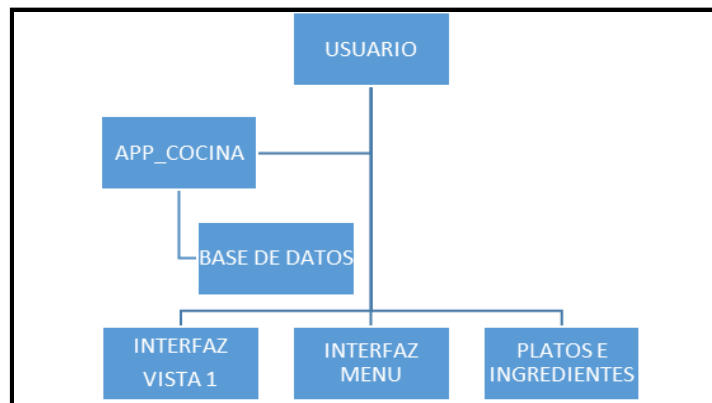


Figura 4: APP\_Cocina

### APP\_STOCK

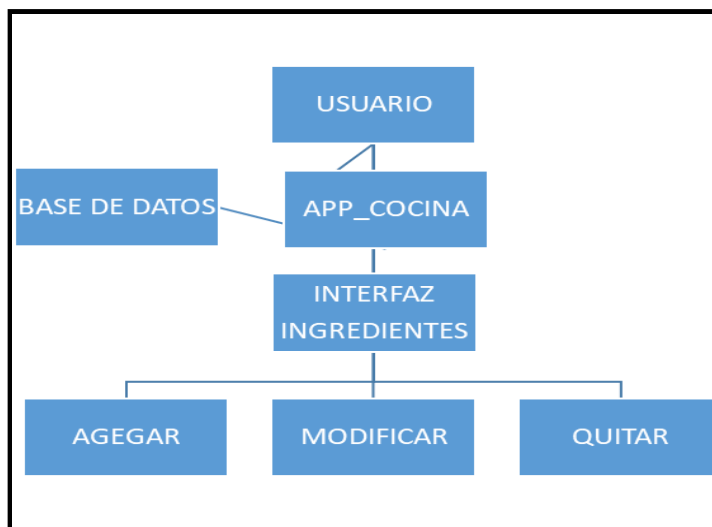


Figura 5: App\_Stock

### **3. Capítulo III: Referentes o marco metodológico**

#### **3.1. Metodología utilizada**

Una de las primeras decisiones que debiésemos tomar y enfrentar cuando comenzamos a enfocar y preparar este proyecto es la elección de la metodología más adecuada para el proyecto SoftCook Management, no es una decisión fácil, ya que existen requerimientos del proyecto de los cuales deben ser debatidos por el equipo de proyectos, cada proyecto posee características diferentes de gestión, entornos variados asociado al propio cliente.

Las metodologías, más utilizadas en los desarrollos de proyectos son **Waterfall o cascada y Agile**, ambas son usables y se pueden implementar en una gran variedad de proyectos.

Para el caso de metodología Agile, esta posee un enfoque iterativo y basado en equipos para el desarrollo, este se enfatiza en la entrega rápida de una aplicación en componentes funcionales completos, estos entregables son priorizados de acuerdo la necesidad que posea nuestro cliente, es por esta razón que nos hemos basado en la metodología Waterfall, ya que proporcionará una aproximación lineal al desarrollo de la aplicación, esta secuencia que sigue este método estará compuesto por las siguientes fases:

- ✓ Captura y documentación de requisitos
- ✓ Diseño
- ✓ Desarrollo de la aplicación
- ✓ Pruebas
- ✓ Corrección de errores
- ✓ Paso a Producción

Cada una de estas etapas deben darse por concluidas antes de comenzar con la siguiente, los requisitos deben ser validados por el cliente antes de avanzar a la fase del diseño.

La metodología por utilizar es mediante el desarrollo de un software que pueda generar interacción entre una base de datos de tal manera que la comunicación efectiva de este con los otros integrantes de la aplicación resuelva

la capacidad de pedidos y platos a realizar. Así tenemos un stock actualizado y ordenado. Dentro de la metodología tenemos los siguientes que mencionamos a continuación.

### 3.1.1. Diagramas del Proyecto - Metodología UML

- UML App\_Cocina

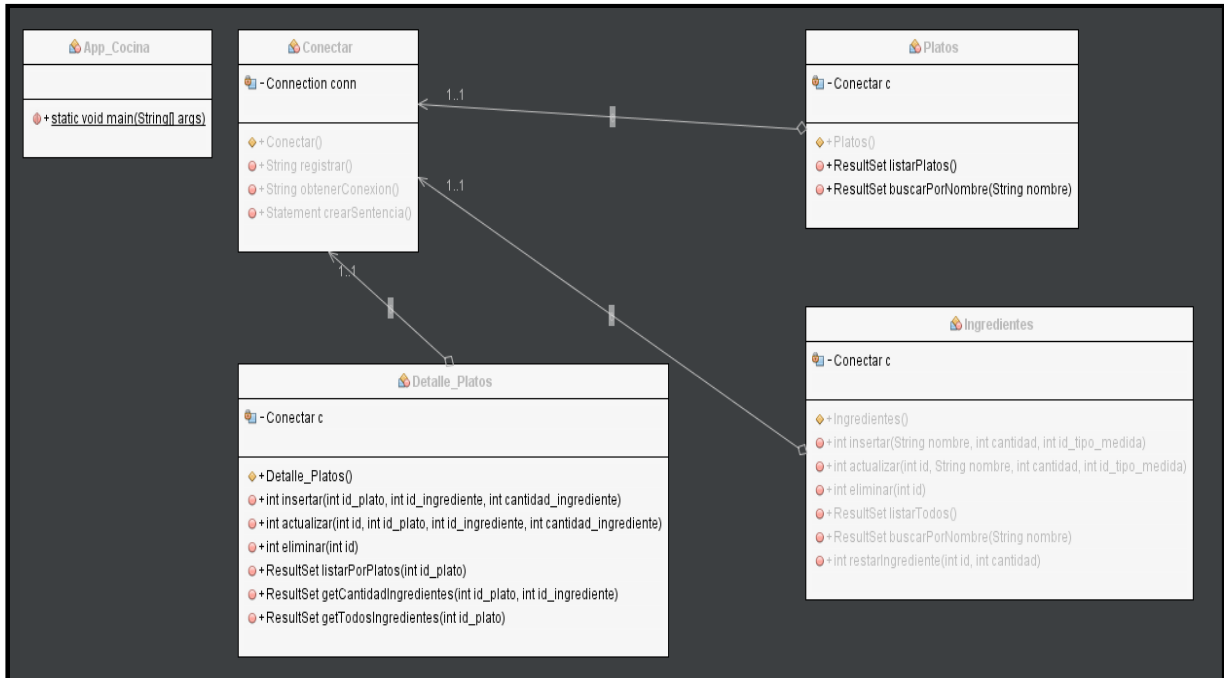


Figura 6: Diagrama UML App\_Cocina

- UML App\_Stock

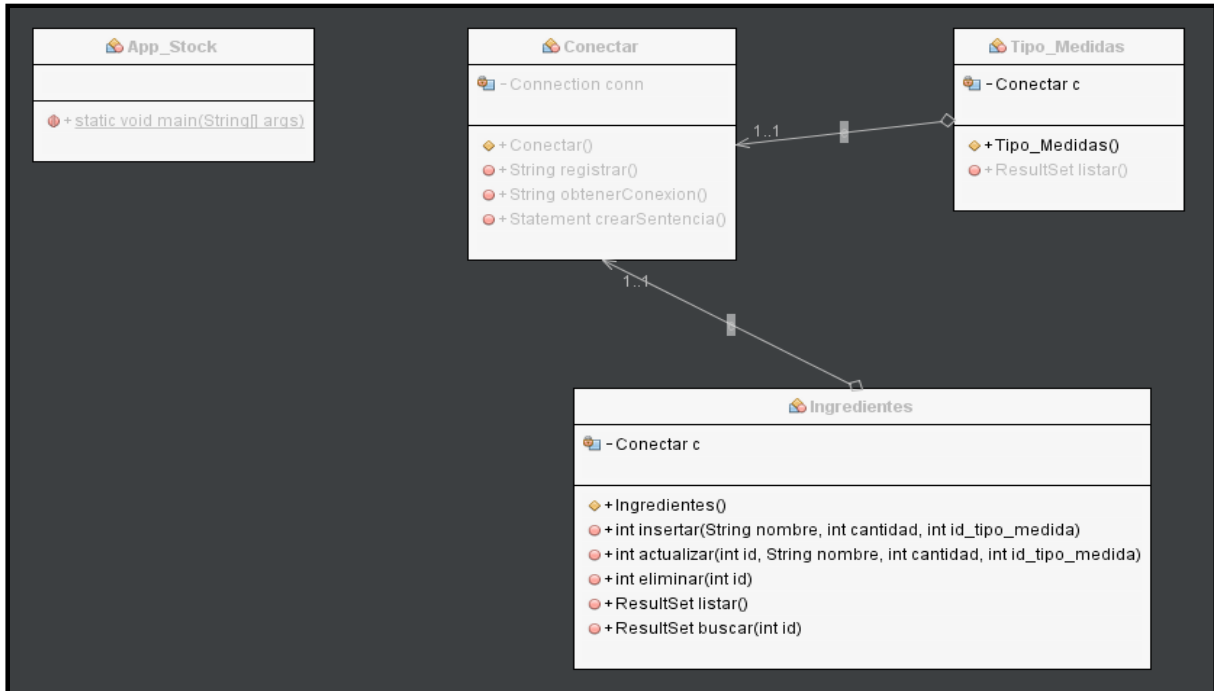


Figura 7: Diagrama UML App\_Stock

### 3.1.2. Diagramas de Caso de uso

**Caso 1:** Cocinero desea preparar plato y este revisa el stock de los ingredientes.

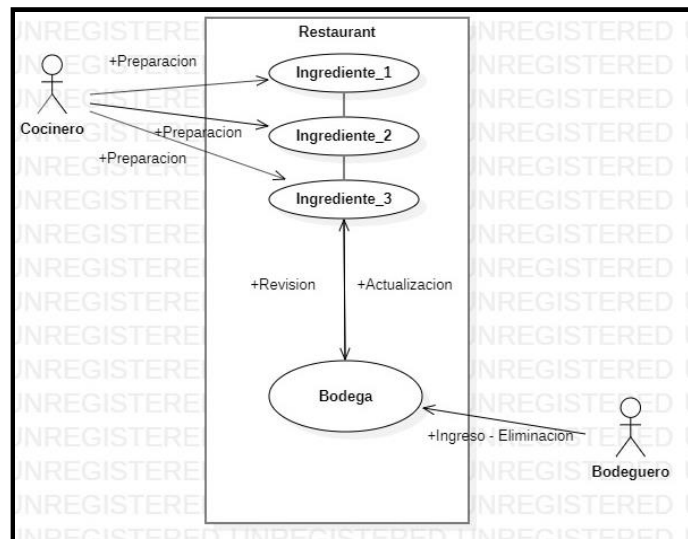


Figura 8: Diagrama Caso de Uso – Caso 1

**Caso 2:** Donde Cocinero tiene que preparar un plato y uno o más ingredientes no posee stock en bodega.

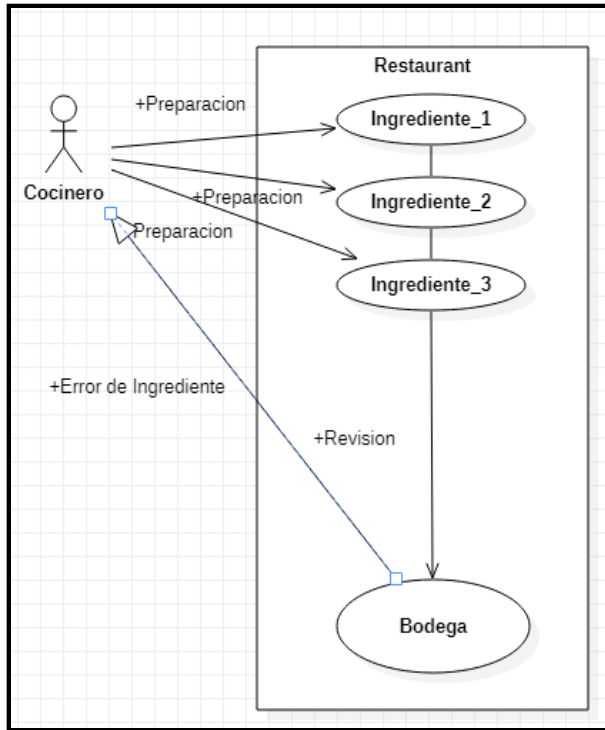


Figura 9: Diagrama Caso de Uso – Caso 2

**Caso 3:** Actualización de Stock de ingredientes en bodega por parte del Bodeguero.

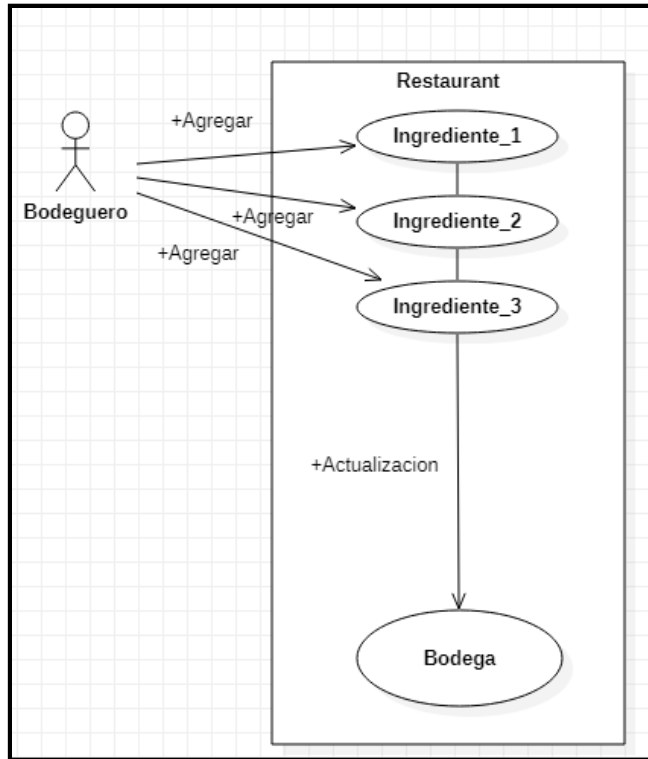


Figura 10: Diagrama Caso de Uso – Caso 3

### 3.1.3. Diagramas de Secuencias

- **Secuencia de pedido desde cocina:**

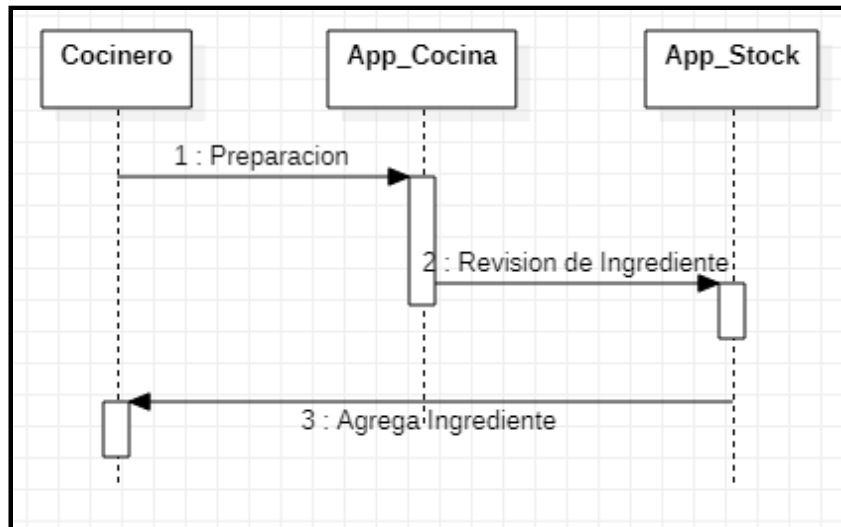


Figura 11: Diagrama Secuencia 1

- **Secuencia de pedido desde cocina con falta de ingrediente:**

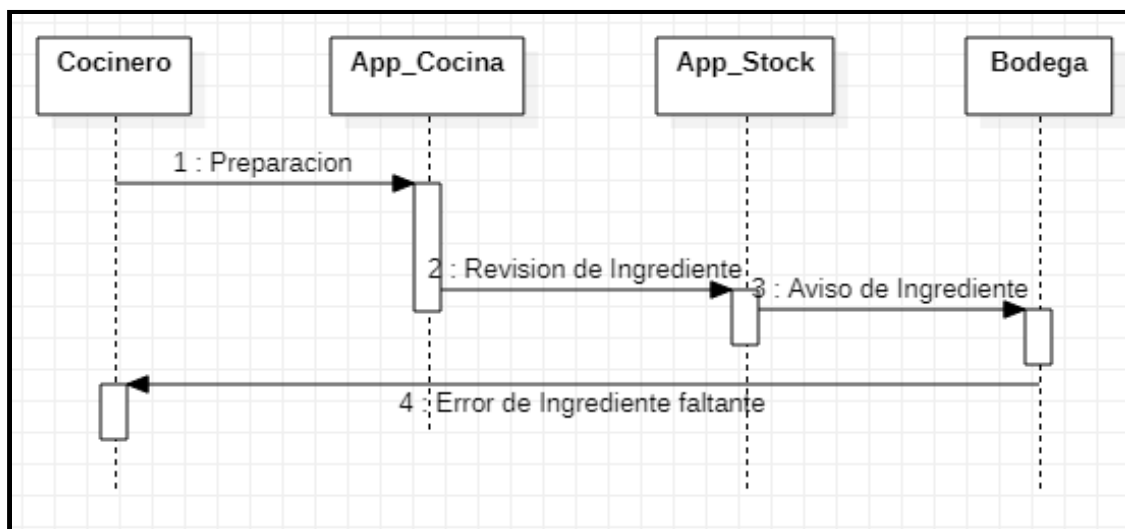


Figura 12: Diagrama Secuencia 2

- **Secuencia de actualización de Bodega por lado de App\_Stock:**



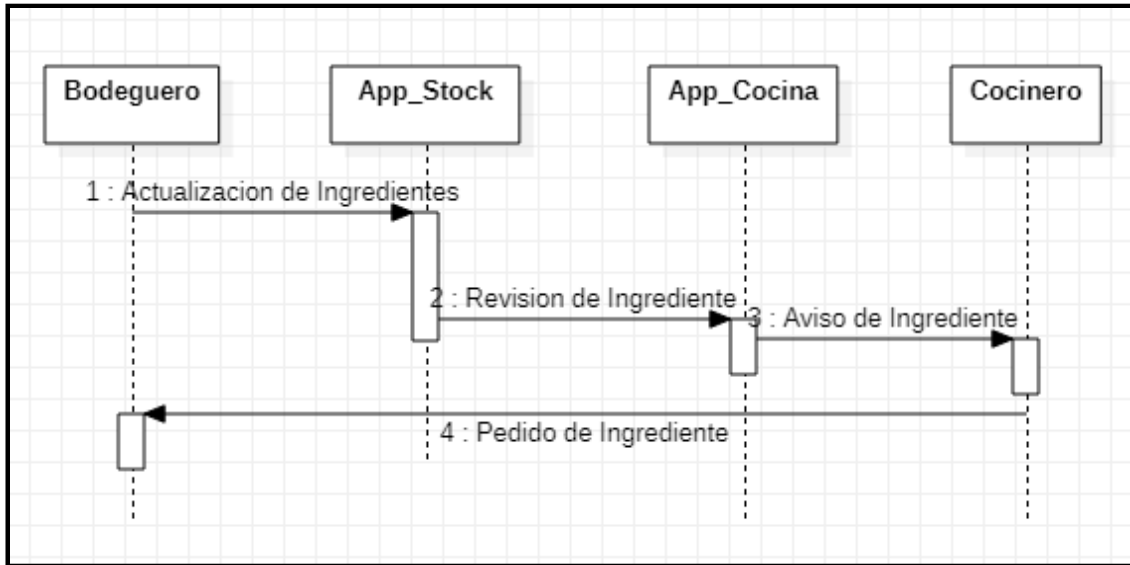


Figura 13: Diagrama Secuencia 3

### 3.2. Beneficios de la metodología Waterfall

<b>Disponibilidad del cliente</b>	✓ Se cuenta un cliente disponible desde el comienzo del proyecto
<b>Velocidad</b>	✓ Se persigue un alcance completo de requisitos, priorizando sobre la velocidad en la entrega del valor al cliente
<b>Alcance</b>	✓ El alcance es perfectamente conocido desde el principio del proyecto.
<b>Priorización</b>	✓ Se asegurará de que el cliente obtendrá todo lo que solicitó desde un comienzo
<b>Presupuesto</b>	✓ Precio Cerrado, si desean cambios hay que activarlos en otro proceso denominado Control de Cambios, del cual generará un aumento presupuestario dependiendo de la solicitud.

Tabla 7: Beneficios metodología

### 3.3. Metodología desarrollo de software

- **Proyecto SoftCook Management:** Conexión de la base de datos MySQL<sup>1</sup> con Netbeans<sup>2</sup> 8.2.

```

18 }
19
20 public String registrar() {
21     try{
22         Class.forName("comm.mysql.jdbc.Driver");
23         return "registrado";
24     }
25     catch(ClassNotFoundException ex){
26         return "No registrado";
27     }
28 }
29
30 public String obtenerConexion() {
31     try{
32         conn = DriverManager.getConnection("jdbc:mysql://localhost/cocina","root","");
33         return "exito";
34     }
35     catch(SQLException e){
36         return "No exito";
37     }
38 }
39
40 public Statement crearSentencia() {
41     try{
42         return conn.createStatement();
43     }
44     catch(Exception e){
45         return null;
46     }
47 }

```

Figura 14: Conexión Base de Datos

- Librería MySQL JDBC Driver (Driver de conexión con MySQL)

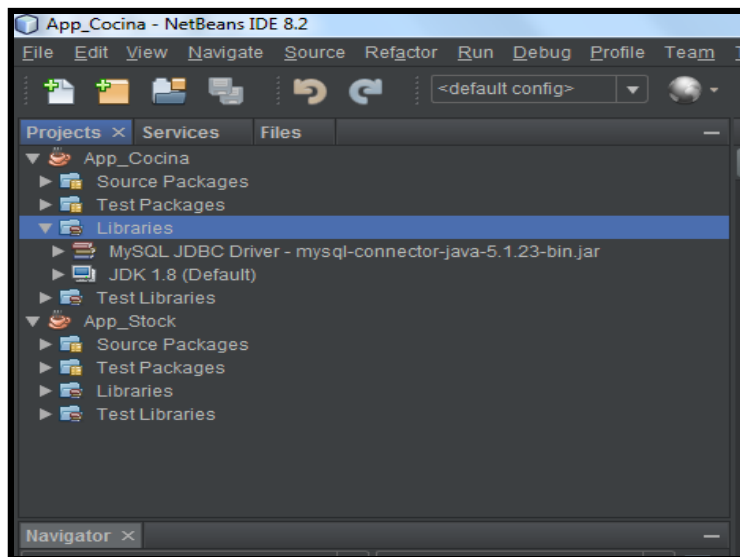


Figura 15: Librería MySQL

- Ejecución de XAMPP<sup>3</sup> y arranque de Apache<sup>4</sup> y MySQL

<sup>1</sup> MySQL: Sistema de gestión de base de datos relacional.

<sup>2</sup> Netbeans: Entorno de desarrollo integrado libre, hecho principalmente para el lenguaje Java.

<sup>3</sup> XAMPP: Paquete de software libre, consiste en el sistema de gestión de base de datos MySQL.

<sup>4</sup> APACHE: Servidor web HTTP de código abierto.

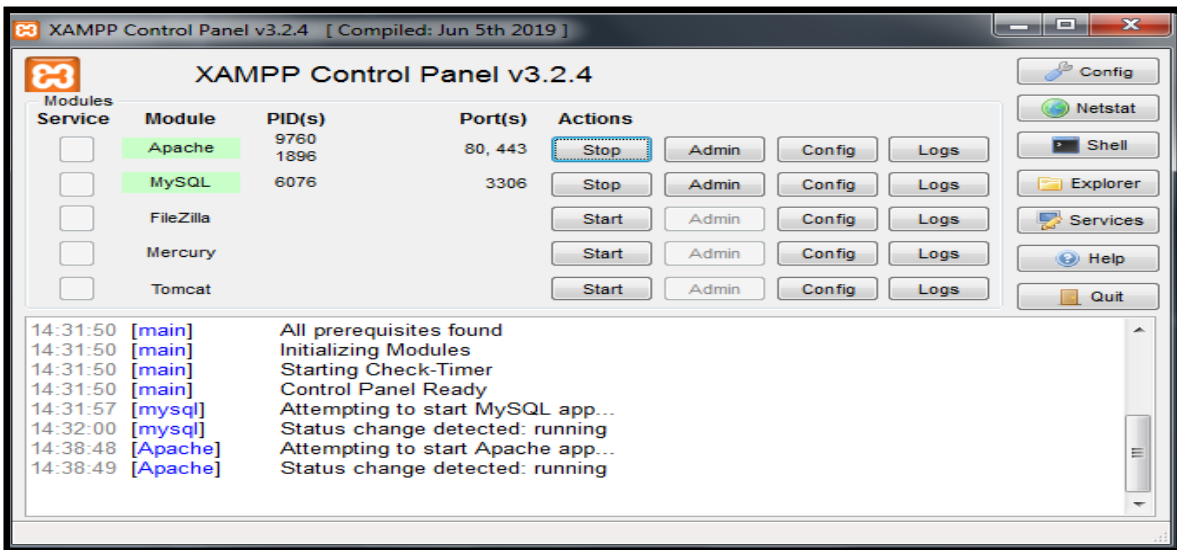


Figura 16: Ejecución XAMPP

- Ingreso a phpMyAdmin<sup>5</sup> y las tablas creadas para la conexión con los programas realizados en Netbeans.

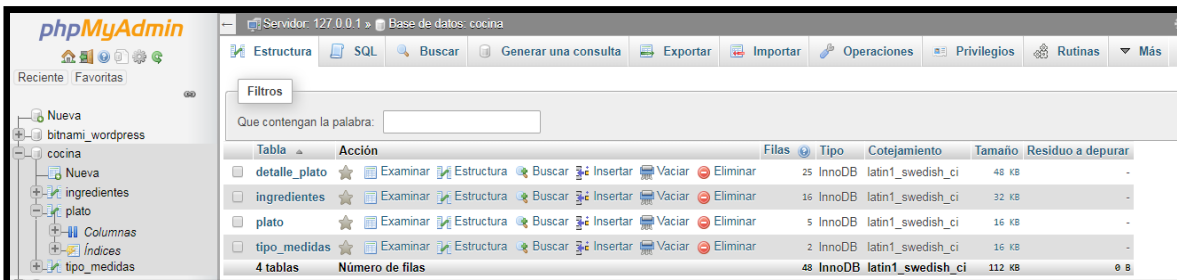


Figura 17: Ingreso PhpMyAdmin

- Ejecución de APP\_Cocina donde seleccionamos los Platos y los ingredientes necesarios.

<sup>5</sup> PhpMyAdmin: Herramienta escrita en lenguaje PHP, con el objetivo de manejar la administración de MySQL.

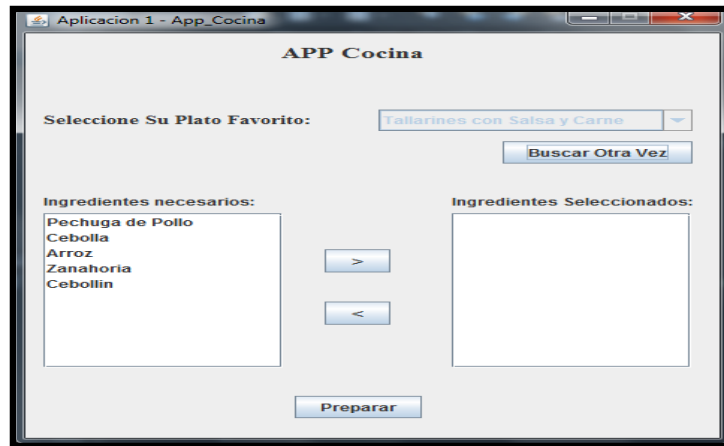


Figura 18: Ejecución APP\_Cocina - Selección

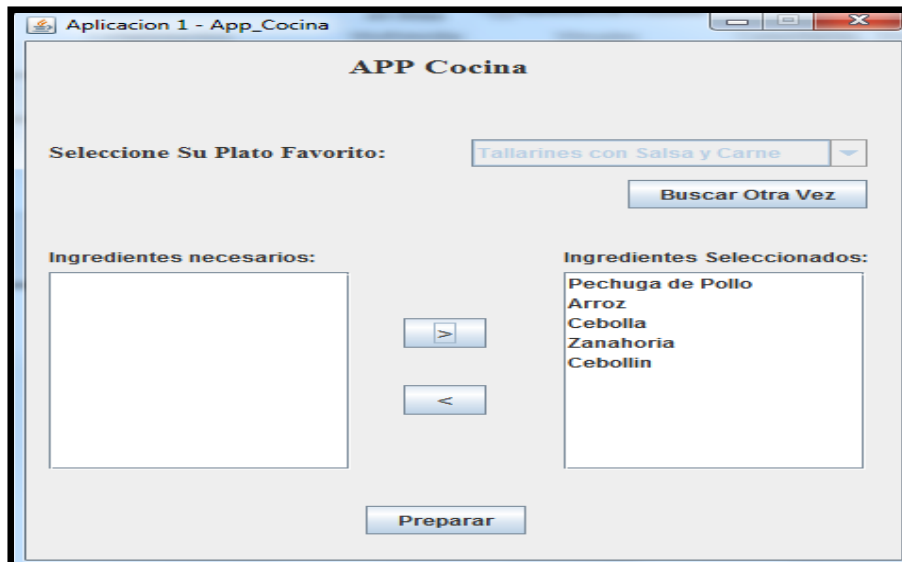


Figura 19: Ejecución APP\_Cocina - Ingredientes

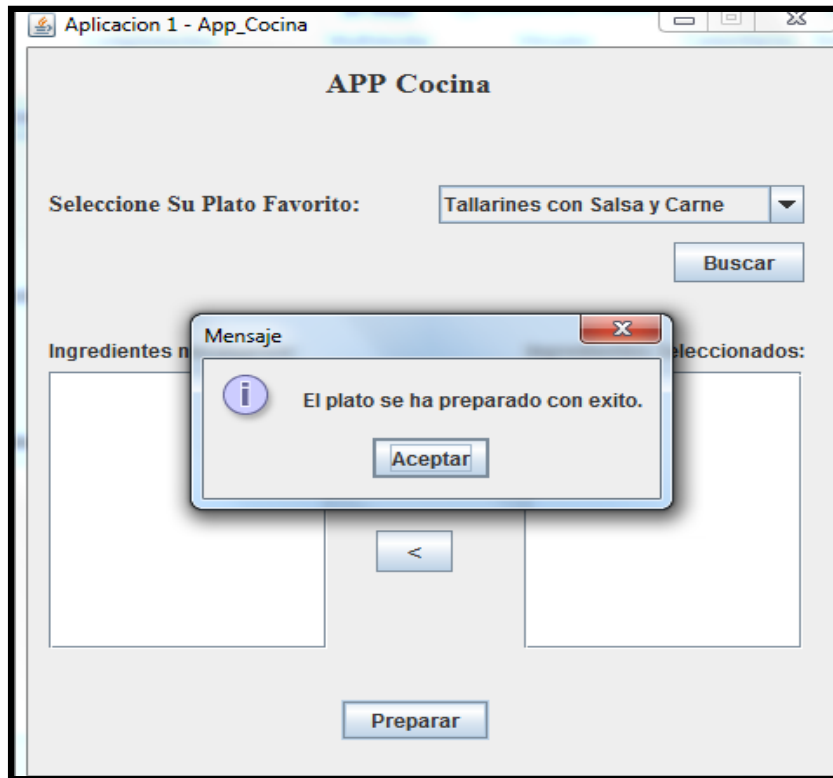
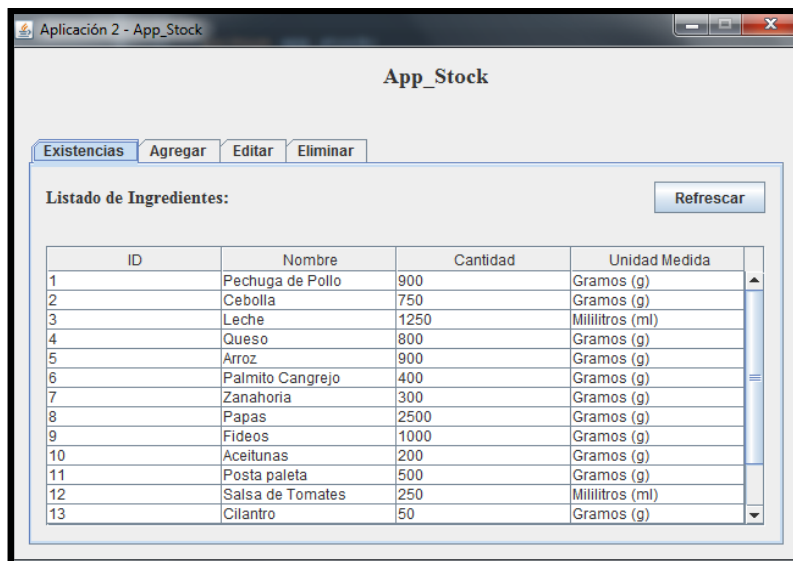


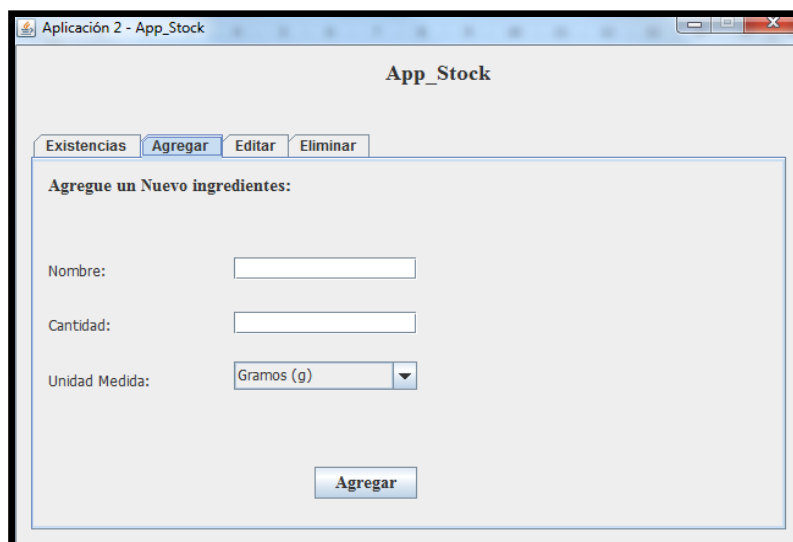
Figura 20: Ejecución APP\_Cocina Mensaje

- Ejecución de App\_Stock, acá revisamos los ingredientes actuales y podemos agregar, editar y eliminar Ingredientes.



ID	Nombre	Cantidad	Unidad Medida
1	Pechuga de Pollo	900	Gramos (g)
2	Cebolla	750	Gramos (g)
3	Leche	1250	Mililitros (ml)
4	Queso	800	Gramos (g)
5	Arroz	900	Gramos (g)
6	Palmito Cangrejo	400	Gramos (g)
7	Zanahoria	300	Gramos (g)
8	Papas	2500	Gramos (g)
9	Fideos	1000	Gramos (g)
10	Aceitunas	200	Gramos (g)
11	Posta paleta	500	Gramos (g)
12	Salsa de Tomates	250	Mililitros (ml)
13	Cilantro	50	Gramos (g)

Figura 21: Ejecución App\_Stock - Existencias



Agregue un Nuevo ingredientes:

Nombre:

Cantidad:

Unidad Medida:

Figura 22: Ejecución App\_Stock - Agregar

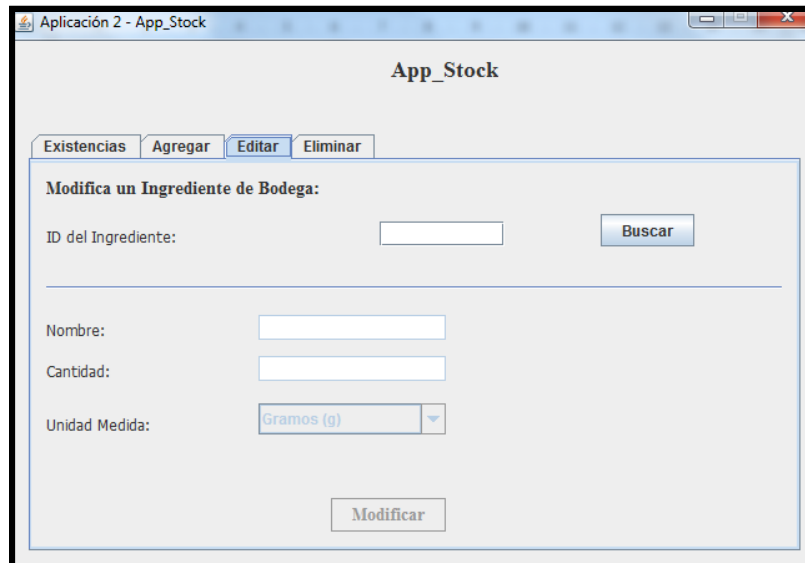


Figura 23: Ejecución App\_Stock - Editar

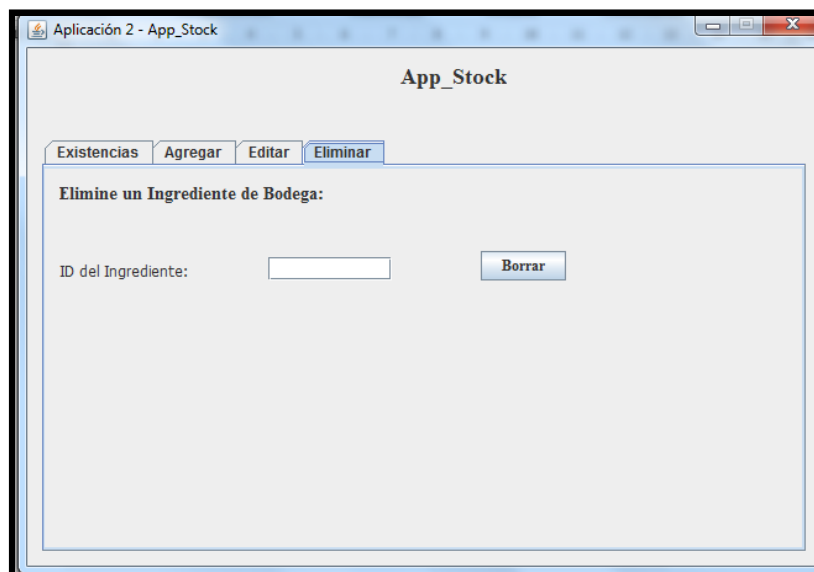


Figura 24 : Ejecución App\_Stock - Eliminar

- Pruebas de integración de aplicaciones App\_cocina y App\_Stock

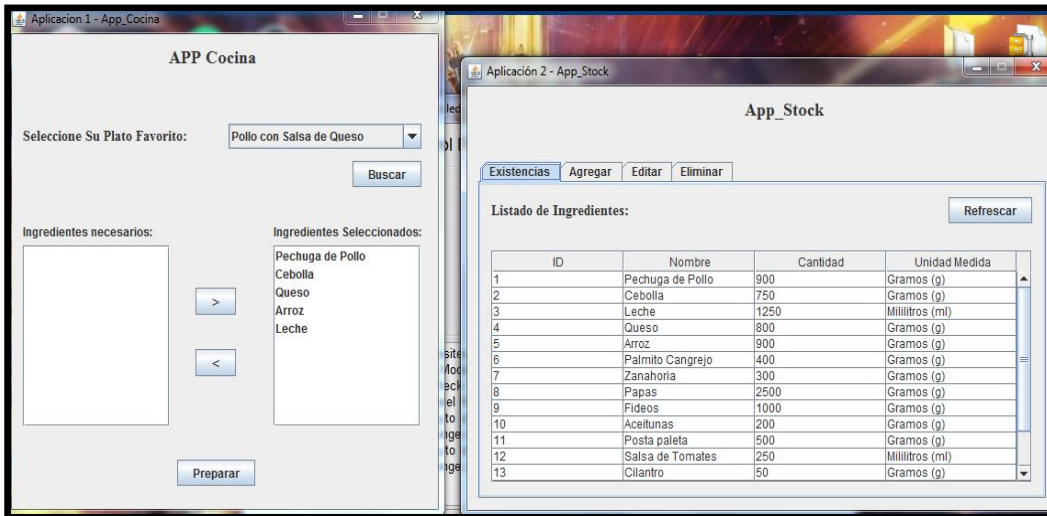


Figura 25: Pruebas Integración

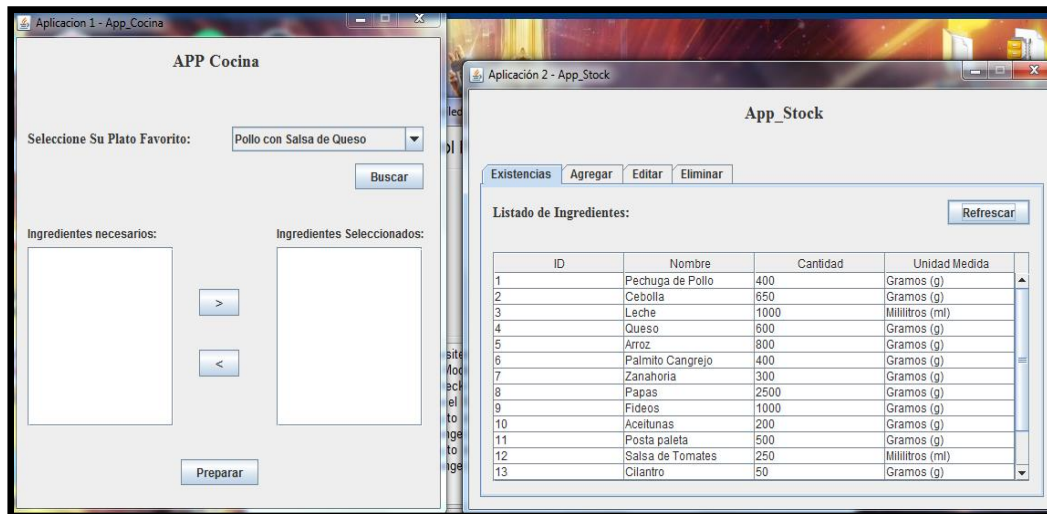


Figura 26: Pruebas Integración

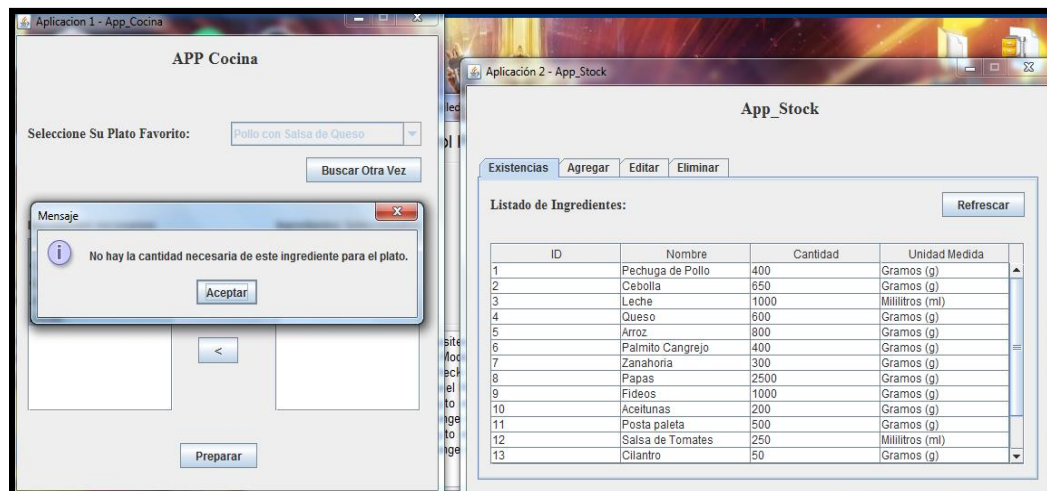




Figura 27: Pruebas Integración - Mensaje

### 3.4. Diccionario de datos

23/3/2020 Vista de impresión - phpMyAdmin 4.9.0.1

**cliente**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
nombre	varchar(30)	No				
apellido	varchar(30)	No				
telefono	int(20)	No				
id	int(20)	No				

**detalle\_plato**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id ( <i>Primaria</i> )	int(11)	No				
id_plato	int(11)	No		plato -> id		
id_ingrediente	int(11)	No		ingrediente -> id		
cantidad_ingrediente	int(11)	No				

Figura 28: Diccionario de Datos – Cliente/Detalle Plato

**Índices**

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	25	A	No	
fk_detalle_plato_plato	BTREE	No	No	id_plato	12	A	No	
fk_detalle_plato_ingrediente	BTREE	No	No	id_ingrediente	25	A	No	

**ingredientes**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id ( <i>Primaria</i> )	int(11)	No				
nombre	varchar(50)	No				
cantidad	int(11)	No				
id_tipo_medida	int(11)	No		tipo_medidas -> id		

**Índices**

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	12	A	No	
fk_ingredientes_tipo_medidas	BTREE	No	No	id_tipo_medida	4	A	No	

Figura 29: Diccionario de Datos – Ingredientes

<b>plato</b>						
Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id ( <i>Primaria</i> )	int(11)	No				
nombre	varchar(60)	No				

<b>Índices</b>								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	6	A	No	

<b>tipo_medidas</b>						
Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id ( <i>Primaria</i> )	int(11)	No				

Figura 30: Diccionario de Datos – Plato

nombre	varchar(20)	No				
--------	-------------	----	--	--	--	--

<b>Índices</b>								
Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	2	A	No	

Figura 31: Diccionario de Datos – Índices

## 4. Capítulo IV: Planificación del proyecto

### 4.1. Cronograma de actividades

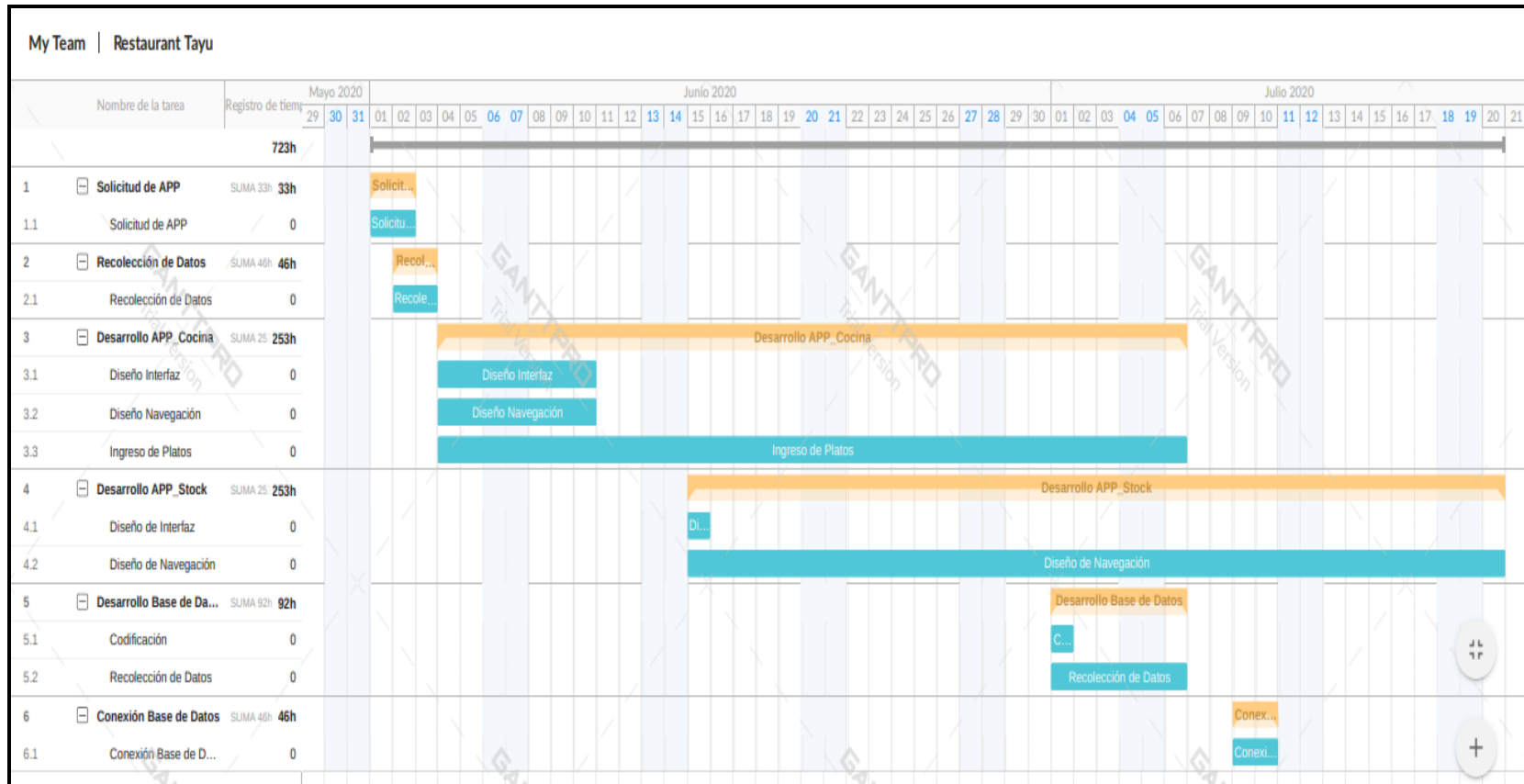


Figura 32 : Carta Gantt

## **4.2. Análisis FODA ¿cuáles son las fortalezas, debilidades, oportunidades y amenazas de su proyecto?**

### **F.O.D.A**

Después de haber analizado el mercado, los usuarios, las empresas y los sistemas que son capaces de desafiar al software creado, a los cuales se enfrentará nuestro sistema, y con el objetivo de tener resultado favorables ante la ejecución e interacción con el usuario, es por tal motivo que se ha llegado al análisis de las fortalezas, oportunidades, debilidades y amenazas las cuales enfrentará SoftCook Management.

#### **4.2.1. Fortalezas**

- ✓ SoftCook Management se inserta dentro del mercado de organización de una empresa, lo cual es necesario dentro de toda corporación, en este caso restaurant, el cual trabaje con stock de productos masivos.
- ✓ La simplicidad de la aplicación desarrollada es crucial para el manejo rápido de la navegación, considerando que la implementación del sistema será dentro una organización de continua demanda.
- ✓ La programación a través de Java y MySQL permite que la aplicación sea compatible en los distintos sistemas operativos, los cuales son más conocidos y de fácil acceso al común del tipo de usuario a tratar.
- ✓ Buena estructura del sistema TI, tanto en codificación como en diseño de navegación, licencias y base de datos.

#### **4.2.2. Debilidades**

- ✓ No existe planificación a mediano y largo plazo.
- ✓ No hay área de proyectos para darle seguimiento al mismo
- ✓ Dado a no ser un sistema pionero dentro del rubro de bodega, SoftCook Management se convierte en una aplicación común para el usuario.

### **4.2.3. Oportunidades**

- ✓ Dado a que la aplicación no está sujeta a empresas, esta puede asociarse a corporaciones (restaurantes) que requieran el software tanto nacionales como internacionales.
- ✓ Es un sistema atractivo para el campo de organización de bodegas y stock dentro de restaurantes.
- ✓ Dado a la fuerte demanda de servicios que “ayuden” al usuario, SoftCook Management se convierte en una buena opción para el mercado.

### **4.2.4. Amenazas**

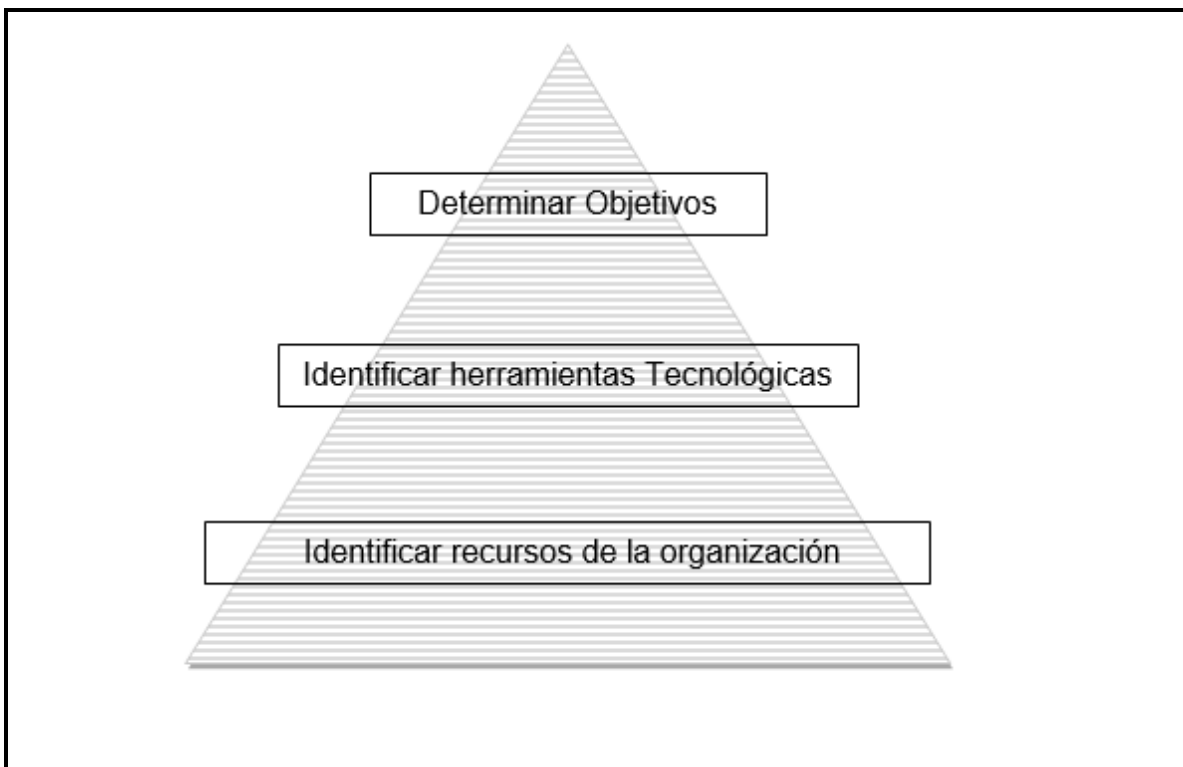
- ✓ Dentro del rubro de organización de bodegas existe una gran gama de empresas conocidas que ofrecen el servicio con paquetes completos.
- ✓ Al ser un sistema poco conocido en el mercado SoftCook Management, el usuario puede mudar sus preferencias a una empresa de conocimiento mundial o nacional.
- ✓ La simplicidad del sistema puede provocar el plagio y mejoramiento de este desde corporaciones significativas o reconocidas en el mercado.
- ✓ El nombre poco conocido de la aplicación convierte en amenaza la propagación comercial del mismo.

## **4.3. Estudio de Factibilidad**

### **Alcance:**

Es necesario analizar los objetivos de la organización para determinar la aplicabilidad de un proyecto que permita el alcance de las metas organizacionales, es por lo que este estudio permite la utilización de diversas herramientas que ayuden a determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la institución.

En esta búsqueda es necesario tomar en cuenta los recursos que disponen la organización o aquellos materiales que el restaurant pueda proporcionar. Y para ello se expondrán los distintos estudios de factibilidad que engloban el correcto desarrollo del sistema.



*Figura 33: Estudio de factibilidad – Alcance*

#### **4.3.1. Factibilidad Técnica**

- **Aplicación 1 (App\_Cocina)**

Esta aplicación será desarrollada bajo el lenguaje de programación JAVA, el cual es un lenguaje de programación que permite trabajar de una forma segura y dinámica, además, presenta la capacidad de ejecutar múltiples tareas en paralelo, pero su principal característica es la independencia de plataforma, ya que implica que un programa desarrollado con este lenguaje puede ejecutarse en diferentes sistemas operativos, sin la necesidad de compilar el código del programa.

Esta aplicación desplegará un menú con diferentes platos, que al seleccionar alguno de ellos, se mostrará una lista de cada ingrediente que se utilizará. A su vez, al momento de seleccionar un ingrediente, este será descontado en forma automática de la base de datos.

Por otra parte, se contempla un stock de seguridad que alerte antes de que ya no quede un ingrediente en particular.

- **Aplicación 2 (App\_Stock)**

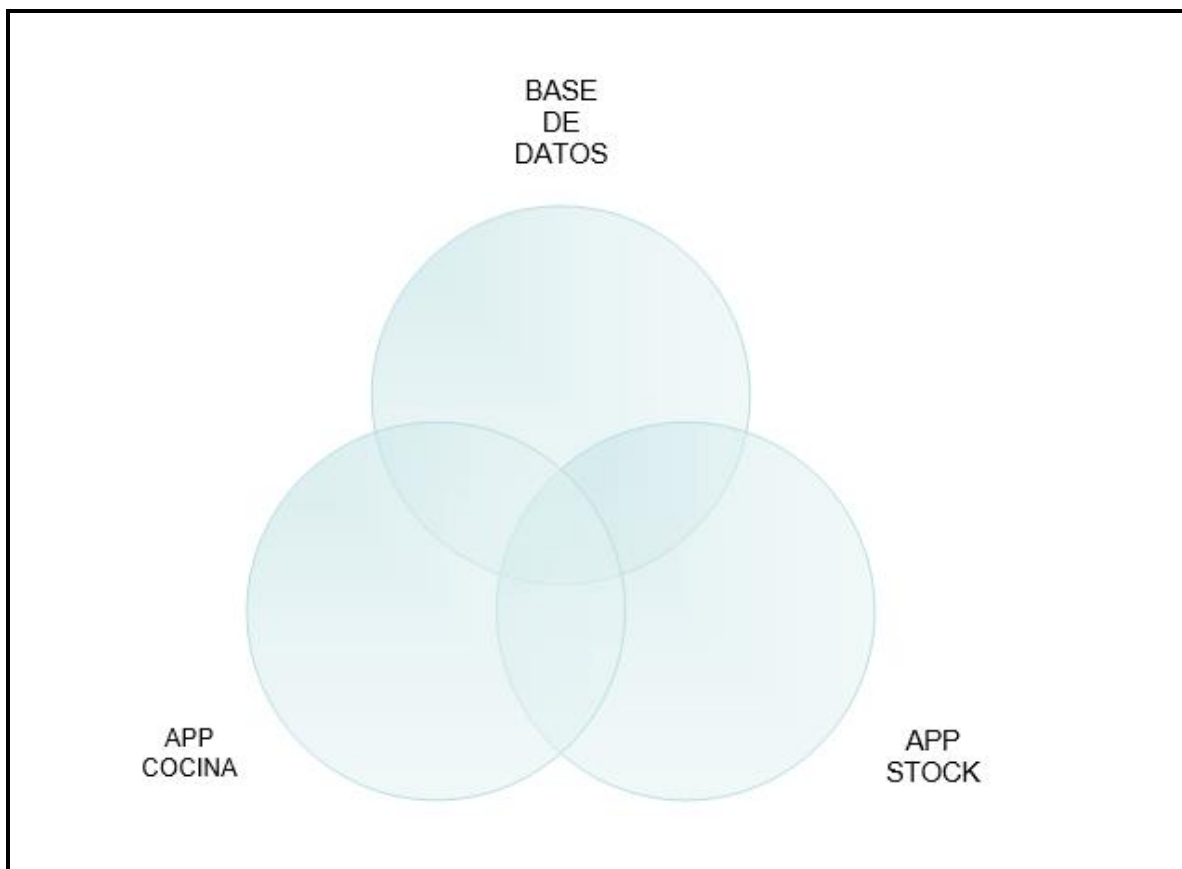
Esta aplicación, al igual que la anterior será desarrollada en el lenguaje de programación JAVA, la cual entregará el stock real de cada ingrediente para la preparación de los diferentes platos, además, deberá contar con un menú mantenedor del sistema, es decir, este menú deberá permitir agregar los ingredientes, modificarlos y eliminarlos.

- **Base de Datos**

El motor de bases de datos elegido fue MySQL para las pruebas, un sistema de gestión de base de datos simple para el desarrollo del proyecto de integración entre dos aplicaciones dentro de una misma base de datos. Se instaló mediante XAMPP para una comprobación local.

A modo de comparación, también se puede gestionar la migración a la base de datos Oracle la cual es considerada como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Además, soporta todas las funciones que se esperan de un servidor robusto, siendo un lenguaje de diseño muy completo, que permite implementar diseños activos, con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente, y por último puede ejecutarse en todas las plataformas. Todo esto es lo que nos llevó a seleccionarla, ya que, además de almacenar una gran cantidad de productos, debe entregar consultas en tiempo real, como es el stock.



*Figura 34: Esquema Base de Datos*

#### **4.3.2. Factibilidad Económica**

Con este estudio se determinará el presupuesto de costos de los recursos técnicos, humanos y materiales, tanto en el desarrollo como en la implementación del proyecto.



Además, contribuirá en la realización del análisis costo-beneficio del proyecto, el cual, a su vez, nos permitirá determinar si es factible de desarrollar.

A continuación, se detallan los costos de los recursos necesarios para el desarrollo e implementación.

<b>Costo de Personal</b>				
<b>Cantidad</b>	<b>Cargo</b>	<b>Valor por HH</b>	<b>Tiempo Estimado</b>	<b>Total</b>
1	Analista	\$ 3.480	723 Hrs.	<b>\$ 2.516.040</b>
1	Programador	\$ 3.120	723 Hrs.	<b>\$ 2.255.760</b>
1	Diseñador	\$ 2.436	723 Hrs.	<b>\$ 1.761.228</b>
1	QA	\$ 1.444	723 Hrs.	<b>\$ 1.044.012</b>

*Tabla 8: Factibilidad Económica – Costo de Personal*

<b>Costo de Mantención</b>			
<b>Horas Mensuales</b>	<b>Horas Semanales</b>	<b>Valor por HH</b>	<b>Total Mensual</b>
20	5	\$ 3.120	\$ 62.400

*Tabla 9: Factibilidad Económica – Costo de Mantención*

<b>Costos Totales</b>	
<b>Items</b>	<b>Valor</b>
Costo de Personal	\$ 7.577.040
Costo de Mantención	\$ 62.400

Costo de Hardware	Sin Costo
Costo de Software	Sin Costo
<b>Total</b>	<b>\$ 7.639.440</b>

Tabla 10: Factibilidad Económica – Costo totales

Los ítems costo de hardware y software no registran valores, debido a que el cliente cuenta con el hardware y software necesario para la realización del proyecto, solo se tendría que revisar las licencias de los computadores. Además, los softwares de desarrollo poseen licencia GNU, lo que favorece a que no existan costos asociados a estos ítem.

En relación a la mantención, ésta quedará estipulada en un contrato que contemplará un total de 20 horas mensuales, las cuales se dividirán en cinco horas semanales y antes de cada fecha de mantención, previamente establecida con el cliente, se realizará una reunión con él o los usuarios para especificar los puntos o aspectos donde se deban centrar las labores. Este contrato se realizará una vez implementado el sistema y las horas que no se ocupen dentro del mes, pasaran al mes siguiente.

Por último, señalar que el costo total asociado a la implementación del sistema es de **\$ 7.639.440 pesos**, donde se realizó un estudio de mercado en relación a los sueldos actuales para cada uno de los profesionales que conforman el equipo de trabajo, para así no escapar de la realidad en el cobro de un sistema de estas dimensiones. Además, con estos resultados obtenidos podemos decir con certeza que el proyecto es viable de realizar.

#### 4.3.3. Factibilidad Legal

Dentro de este ámbito, se pueden mencionar los siguientes puntos importantes:

- **Implementación de software:**

La implementación del software que permitirá, tanto el listado de platos y selección de ingredientes, como el control de stock, será desarrollado mediante software libre con licencia GNU<sup>6</sup>, que el caso de JAVA.

- **Implementación de la base de datos:**

Para la implementación de la base de datos, el motor de bases de datos elegido fue MariaDB (MySQL) el cual está dentro del paquete de instalación XAMPP, para la realización de las funcionalidades de la base de datos y las dos aplicaciones se creó una conexión utilizando el servidor local que implementa XAMPP. Se tomó como opción la base MySQL, Un sistema de bases de datos relacional desarrollado bajo licencia dual, Licencia pública general/Licencia comercial por Oracle Corporation, considerada como la base de datos de código abierto más popular del mundo, sobre todo para entornos de desarrollo web.

Permite que aplicaciones escritas en diversos lenguajes de programación tengan acceso a base de datos MySQL, otra de sus características es que es muy rápida en la lectura, lo que nos llevará a seleccionarla.

Se revisó otras alternativas de base de datos donde la utilización de Oracle Database SE<sup>7</sup>, pero como base de datos diseñada para el almacenamiento de grandes cantidades de información esta nos hacía consumir recursos del sistema el cual no ayudaba a la generación del proyecto tal como se ha mencionado dentro del proyecto.

Para clarificar que hoy MariaDB viene con XAMPP así como antiguamente con MySQL. Pero al partir de la compra de MySQL por Oracle hoy su licencia es dual tal como se menciona anteriormente. Hoy la diferencia de MariaDB con MySQL solo es mejoras a nivel de velocidad, almacenamiento entre otras mejoras.

#### **4.3.4. Factibilidad de Recursos**

A continuación, se describirá los recursos, tanto materiales y humanos que se necesitarán para la ejecución del proyecto.

---

<sup>6</sup> GNU: Sistema operativo del tipo UNIX, que posee la particularidad de ser software libre.

<sup>7</sup> Oracle Database SE: Sistema de gestión de base de datos de tipo objeto-relacional.

Debido a que se trata de un sistema promedio, con costos acorde al mercado, la utilización de equipos potentes es innecesaria, motivo por el cual la utilización de equipos estándar es adecuada para el sistema.

Por otra parte, para cumplir satisfactoriamente con la ejecución del sistema se conformará un equipo de trabajo de cuatro profesionales, que permitirá alcanzar los objetivos establecidos y satisfacer los requerimientos del cliente, donde cada profesional desempeñará uno o más roles, de acuerdo a las necesidades del proyecto.

A continuación, se muestra una tabla con la explicación antes mencionada:

<b>RECURSOS PARA EL DESARROLLO DEL PROYECTO</b>			
<b>Tipo de Recurso</b>	<b>Nombre del Recurso</b>	<b>Descripción</b>	<b>Cantidad</b>
Recursos Humanos	Experto en el Área de Desarrollo	Analista	1
		Programador	1
		Diseñador	1
		QA	1
Hardware	PC	Procesador 2.0 GHz 4 Gb de RAM 500 Gb disco duro S.O Windows/MAC OS	2
Software	MySQL	Gestor de base de datos	1
	NetBeans	Entorno de desarrollo IDE	1

*Tabla 11: Factibilidad de Recursos – Desarrollo del Proyecto*

#### **4.3.5. Factibilidad de Mercado**

- **Población Objetivo:**

Hoy en día en Chile, el mercado gastronómico ha venido creciendo exponencialmente en promedio, debido a diversos factores, entre ellos el aumento de la oferta gastronómica.

Las tendencias han llevado a la creación de ciertos espacios gastronómicos donde se ofrecen distintas especialidades, los cuales, en los últimos años, han impulsado al crecimiento de la industria, lo que lleva a generar una situación provechosa sumado al demandante mercado de sistemas informáticos a la medida, por lo que, con nuestro sistema, es posible atacar el centro de la problemática y lograr insertarnos en dicho mercado.

Dentro de la factibilidad del mercado se podrá encontrar los distintos tipos de mercado los cuales se deben enfrentar con alternativas eficientes para lograr el éxito del proyecto SoftCook Management.

- **Mercado consumidor:**

Nuestro proyecto se centrará en el sector gastronómico, específicamente en un restaurant ubicado en Villarrica, donde se detectó una problemática en la administración de los ingredientes de los platos y el control de stock.

- **Análisis de la demanda:**

Teniendo en cuenta que el servicio que se propone ya se encuentra en el mercado, que corresponde a la categoría de desarrollo de software a la medida, estaría diferenciado por las problemáticas abordadas (administración de ingredientes y control de stock), ya que los sistemas existentes, mayoritariamente abordan la realización de pedidos por parte de los clientes. Además, al ser un servicio acorde al mercado que presenta una competencia directa, crea una posibilidad de negocio e inserción en el mercado muy auspiciosa.

Como bien sabemos el software se desarrolla y no se fabrica, esto puede significar que se puede empaquetar y comercializar como producto estándar, con lo que se deduce que el desarrollo de un programa requiere como mínimo de 12 a 18 meses.

Con el avance tecnológico y aumento de la automatización de los procesos, el desarrollo de software a la medida pronostica un futuro auspicioso, ofreciendo al mercado soluciones innovadoras para la integración de la información en nuevos sectores no explorados, asegurando el éxito en este campo.

- **Mercado competidor:**

Este mercado está formado por distintas empresas que producen y venden productos o servicios similares a los que la compañía está ofreciendo.

- **Mercado distribuidor:**

Este mercado se enfoca a la distribución del producto, pero uno de los puntos más importantes de este mercado es el valor, ya que es importante escoger un mercado distribuidor que, en cuanto a costos, pueda competir de buena forma ante el mercado competidor.

#### **4.3.6. Factibilidad Operacional**

El principal propósito de este proyecto es administrar los diferentes ingredientes de los distintos platos que se preparan en el restaurante, donde se mostrará un menú con los distintos platos que al seleccionar un ingrediente se descontará del stock, a su vez el sistema deberá permitir conocer el stock en tiempo real. Conjuntamente a las problemáticas, se suma el desconocimiento de

los recursos informáticos, donde se debe capacitar a los usuarios para un correcto manejo del sistema y lograr una utilización eficiente de este.

	<b>Creación de sistema para la administración de ingredientes y conocimiento de stock</b>	<b>Capacitación al personal para el correcto manejo del sistema.</b>
<b>Tiempo</b>		
1. Corto Plazo 2. Mediano Plazo 3. Largo Plazo	<b>3</b>	<b>3</b>
<b>Beneficiarios</b>		
1. Pocos 2. Suficientes 3. Muchos	<b>2</b>	<b>2</b>
<b>Costos</b>		
1. Corto Plazo 2. Mediano Plazo 3. Largo Plazo	<b>3</b>	<b>3</b>
<b>Impacto</b>		
1. Altos 2. Medios 3. Bajos	<b>3</b>	<b>3</b>
<b>Factibilidad</b>		
1. Poco Probable 2. Medianamente Probable 3. Probable	<b>3</b>	<b>3</b>

*Tabla 12: Factibilidad Operacional*

#### **4.3.7. Factibilidad de Tiempo**

Se ha determinado que no se presentarían inconvenientes con la implementación, diseño y con usuarios, debido a que no se verían afectadas sus actividades por periodos prolongados de tiempo y dado a la simplicidad de la interfaz del sistema, la capacitación de los usuarios solo se realizará una vez finalizada la etapa de pruebas con una previa coordinación.

Tiempos que podemos considerar:

<b>Implementación</b>	<b>Tiempo</b>
Instalación y Configuración del sistema.	3 Hrs. Aproximadas
Configuración de red	2 Hrs. Aproximadas
<b>Ejecución</b>	<b>Tiempo</b>
Capacitación Usuarios y Administrador	3 Hrs. Aproximadas

Tabla 13: Factibilidad de Tiempo

#### 4.3.8. Check List de aseguramiento de calidad

A continuación, se mostrarán los diferentes check list de aseguramiento de calidad, para así asegurar que se cumple con lo especificado en el sistema.

<b>Actividad</b>	<b>SI</b>	<b>NO</b>
¿Están definidos los requerimientos?	X	
¿El Sistema SoftCook Management permitirá el acceso mediante un nombre de usuario y contraseña únicos?		X



¿El sistema SoftCook Management permitirá recuperación de contraseña de usuario?		X
¿El Sistema SoftCook Management emitirá informes o reportes?	X	
¿El Sistema SoftCook Management permitirá llevar el control de ingredientes?	X	

Tabla 14: Check List

#### 4.3.9. Análisis de Riesgos

Se considera cada riesgo identificado y se realiza un juicio acerca de la probabilidad y gravedad de dicho riesgo como se muestra en la siguiente tabla N°14 Valor del Riesgo.

Se asignarán los riesgos en la tabla N°15 Análisis de Riesgo.

#### Valor del Riesgo:

Muy baja	Baja	Moderada	Alta	Muy Alta
< 10%	10% al 25%	25% al 50%	50% al 75%	>75%

Tabla 15: Valores del Riesgo

Los efectos del riesgo pueden estimarse como:

- Catastróficos: Amenazan la supervivencia del proyecto
- Graves: Causarían grandes demoras
- Tolerables: Demoras dentro de la contingencia permitida

#### Análisis del Riesgo:

Riesgo	Definición	Probabilidad	Efectos
Cambio del personal	El personal capacitado y con experiencia abandonará el proyecto	Alta	Catastrófico

	antes que termine		
Falta de recursos de software	Importante en el proyecto, no se entregará a tiempo	Moderada	Grave
Cambio de requerimientos	A futuro pueden repercutir en la calidad del producto a desarrollar	Moderada	Grave
Requerimientos mal definidos	Requerimientos vagos o poco entendibles por las partes vinculadas	Alta	Tolerable
Personal enfermo	Personal puede enfermar en el transcurso del desarrollo del proyecto	Moderada	Grave
Falla de algún componente de Hardware	Es posible que el hardware se dañe al ser afectado por ejemplo con un corte de luz	Moderada	Catastrófico
Diseño inadecuado	El diseño no cumple con la expectativa del restaurant	Moderada	Tolerable
Intento de acceso no autorizado	El sistema no cuenta con ingreso de usuario y contraseña	Alta	Catastrófico

Tabla 16: Análisis del Riesgo

#### 4.4. Cuantificación del Proyecto en esfuerzos y costos

El proyecto SoftCook Management se podrá conceptualizar como un cambio ventajoso en el ámbito de calidad y puede considerarse que es una consecuencia de la puesta en marcha una vez instalado en el restaurant.

La implementación de un proyecto de mejora trae consigo costos y beneficios, siendo un desafío enfrentar la necesidad de valorar esos costos y beneficios de la forma más objetiva posible para apoyar una mejor decisión respecto de implementar o no una iniciativa de mejora.

Los proyectos de mejora también se pueden abordar como un proyecto de inversión porque desde una perspectiva más global se pueden entender como una

intervención en un medio para dar solución a una problemática existente y lograr un cambio deseado. Permite justificar la intervención desde diferentes puntos de vista para dar solución a una problemática, incluso es un enfoque donde se pueden estimar ventajas y desventajas que se derivan en asignar recursos.

- **Ventajas:**

- ✓ Mejora en la satisfacción de los clientes
- ✓ Mejora en la gestión del restaurant
- ✓ Mejora de la eficiencia
- ✓ Mejora en la competitividad y reputación de la imagen del restaurant
- ✓ Permanencia del restaurant en la ubicación establecida
- ✓ Responder a una mejor petición de los clientes
- ✓ Mejoras en el nuevo servicio
- ✓ Disminución de los reclamos de clientes

- **Desventajas:**

- ✓ Costo de la implementación
- ✓ Mala disposición del personal autorizado para el uso del sistema
- ✓ Probabilidad de existir personal poco capacitado o desinteresado en hacer uso del nuevo sistema

#### **4.5. Justificación viable del Proyecto**

En la actualidad es necesario e importante que todas las organizaciones o empresas cuenten con sistemas informáticos para la automatización, tanto de los registros, como del control, permitiendo el manejo de los datos, procesos y reportes que posteriormente las empresas verifican la rentabilidad de estos. La automatización de los sistemas colabora con la necesidad de acceso y conocimiento de la información en cualquier momento.

El cliente que motivó este proyecto, el cual corresponde a un restaurant, no cuenta con un sistema informático que le permita administrar de una manera más efectiva los diferentes ingredientes que se emplean para realizar los distintos platos que confecciona el menú, además, se le suma la carencia de otro sistema

que le permita o proporcione la información correcta del stock, para así poder gestionar la adquisición de los productos faltantes.

La puesta en marcha del sistema SoftCook Management, ayudará a mejorar la administración de los ingredientes y mantener un control más efectivo del stock, evitando la pérdida de la preparación de un plato, lo que posteriormente significaría la pérdida de una venta. Además, garantizará que lo que el cliente solicite se proporcionará en el tiempo y forma esperada.

En relación con la viabilidad económica, hemos realizado el análisis costo-beneficios del proyecto, donde se han calculado los márgenes del beneficio sobre el costo total, mostrando la rentabilidad del proyecto en relación con las horas invertidas y el beneficio que se obtendrá de él.

El software se desarrolla, se instala, no se construye de forma física, como cualquier otro producto clásico. Es por lo que los costos radican principalmente en horas de ingeniería, ya que los softwares de desarrollo que se utilizarán poseen licencia GNU, por lo cual solo se tendría que revisar las licencias de los computadores, los cuales serán proporcionados por el cliente.

En cuanto a la viabilidad técnica, podemos decir que, debido a que se trata de un sistema, la utilización de equipos robustos es innecesaria, bastaría con la utilización de tecnología estándar existente en el campo de la informática, la cual nos permite abordar satisfactoriamente los objetivos y requerimientos establecidos. Por otra parte, el lenguaje empleado para el desarrollo es de código abierto por lo que tampoco presentaría problemas.

Por último, la implementación de SoftCook Management permitirá, no solo ayudar a mejorar la administración de los productos y el control de stock, sino que también presenta una competencia directa, creando una posibilidad de negocio e inserción en el mercado muy auspiciosa, como lo es el desarrollo de software a la medida, que ofrece soluciones innovadoras en nuevos sectores no explorados, asegurando el éxito en este campo.

#### **4.5.1. Justificación de las herramientas de desarrollo utilizadas**

El decidir que herramientas utilizar para el desarrollo del proyecto debemos revisarlo conjuntamente con el análisis del problema, la metodología a utilizar y las factibilidades técnicas y económicas.

Por parte del análisis del problema el resolver la administración de los ingredientes desde la cocina y la bodega es importante ya que estos dos sectores de un restaurante son complementarios uno de otro. Y por esto debemos determinar que una herramienta de desarrollo basada en la portabilidad debía obtener mejores resultados a un mediano y largo plazo.

Dentro de las factibilidades técnicas se debió buscar la mejor alternativa de lenguaje de programación que se integrará con una base de datos como MySQL y por esto se realizó la programación de las dos aplicaciones en Java utilizando el IDE Netbeans 8.2. Para el desarrollo de la base de datos se utilizó Xampp V 3.2.4 el cual posee MySQL incorporado y con el mismo programa se realizará un servidor virtual de la base de datos. Tal como se detalla en el punto 4.3.2.

En relación a la factibilidad económica, donde se determinó los costos asociados a la programación, se llegó a la conclusión de utilizar herramientas Open Source y el lenguaje JAVA, para reducir los costos asociados a estos, además este lenguaje es un sistema robusto y multiplataforma, con una comunidad de programadores muy amplia en el sector lo que no presentaría mayores dificultades en su utilización.

### **Cuadro comparativo de Lenguaje de programación Java**

<b>Ventajas</b>	<b>Desventajas</b>
Multiplataforma	Velocidad
Orientación a objetos	Rendimiento
Robusto	Escalabilidad con tecnología actual
Realización de Pruebas JUnit <sup>8</sup>	Sintaxis engorrosa
Amplio Soporte de la comunidad	Java es dependiente de su máquina virtual JVM

*Tabla 17: Cuadro comparativo*

<sup>8</sup> JUnit: Conjunto de bibliotecas utilizadas para hacer pruebas unitarias con Java.

Como se visualiza en el cuadro comparativo la programación en lenguaje java posee mayores ventajas que desventajas dentro de este proyecto. Aun cuando la Escalabilidad en el tiempo podríamos migrar el tipo de programación y enfocarnos en la realización de tecnologías que hoy se están ejecutando como PWA<sup>9</sup> ya que esta tecnología hace que podamos acceder de forma más rápida y mediante dispositivos móviles las 24 horas.

#### **4.6. Evaluación de alternativas en el mercado**

En el siguiente punto se procederá a analizar las alternativas en el mercado, las cuales se pueden dar a través del análisis de integrar nuevas y distintas tecnologías en base al sistema que se ha creado, para lograr mayor cobertura en el caso de la solicitud de otro tipo de alternativa de funcionamiento y ejecución de SoftCook Management. Tomando en cuenta las ventajas y desventajas que se localizan en la construcción de un software propio, licenciado o externalizado, de las cuales cada una tendrán potencialidad y discrepancia para caracterizar tales alternativas.

##### **4.6.1. Posible Competencias en el mercado actual**

- **Alternativa 1 – “Resto”:** Es un software gastronómico que se adapta a todos los tamaños de negocios (pequeño, mediano y grande), y en todos los formatos, restaurantes, bares, parrillas, pizzerías, heladerías, cafeterías, entre otros. Es un sistema totalmente escalable que acompaña su crecimiento.

##### **Funcionalidades de Resto:**

- ✓ Administra una o múltiples cajas con cierres independientes
- ✓ Selección de productos por código, descripción, sinónimo o código de barra

---

<sup>9</sup> PWA: Progressive Web Apps por sus siglas en inglés, tipo de software de aplicación que se entrega a través de la web.

- ✓ Búsqueda inteligente de artículos
- ✓ Permite imprimir en cocina el pedido de sus clientes
- ✓ Administra inventario en múltiples bodegas
- ✓ Descarga de inventario centralizada, por producto o por sector
- ✓ Múltiples informes y estadísticas de inventario
- ✓ Múltiples destinos de impresión para un mismo pedido
- ✓ Impresión de preferencias (cocido, jugoso, con hielo, etc.)
- ✓ BackOffice completo
- ✓ Ingreso, autorización, actualización y gestión de solicitudes de compras
- ✓ Generación, autorización y seguimiento de órdenes de compra
- ✓ Seguimiento de pendientes de entrega o facturación
- ✓ Contabilización por artículo o concepto
- ✓ Múltiples informes y estadísticas sobre la gestión de compras
- ✓ Ingresos de platos dentro y fuera de menú
- ✓ Liberación automática o manual de mesas
- ✓ Productos habilitados por horario
- ✓ Visualización del último o últimos pedidos
- ✓ Búsqueda de clientes por múltiples criterios
- ✓ Posibilidad de administrar salón y delivery desde una misma pantalla
- ✓ Manejo de reservas y listas de espera
- ✓ Emisión de facturas electrónicas

### **Desventajas**

- ✓ La adquisición del software implica un paquete de funciones que no necesariamente satisface los requisitos de la problemática expuesta por el cliente.
- ✓ Se centra más en la venta y facturación de los platos que en resolver la problemática que origino la creación de SoftCook Management.

- **Alternativa 2 - SmartRestoBar:** Es un software para el rubro gastronómico, específicamente para restaurantes, resto bar o café bar, el cual presenta tres diferentes planes, plan estándar, plan Premium y plan Smart, de acuerdo con el promedio mensual de facturas. Este software está disponible para los sistemas operativos Windows, Linux y Sistema Android.

#### **Funcionalidades de SmartRestoBar:**

- ✓ Control de caja
  - ✓ Cocina Touch
  - ✓ Barra Touch
  - ✓ Delivery
  - ✓ Anfitrión, el cual ayuda a gestionar los turnos de los clientes en estado de espera
  - ✓ Familia de productos (categorías)
  - ✓ Productos genéricos (no elaborados)
  - ✓ Productos preparados (elaborados)
  - ✓ Control de Stock
  - ✓ Administrador de proveedores
  - ✓ Administrador de personal.
  - ✓ Configuración de jornadas.
  - ✓ Configuración de salones y mesas-barras.
  - ✓ Configuración de dependencias.
  - ✓ Configuración datos de empresas.
  - ✓ Estadísticas.
  - ✓ Administrador de cliente (delivery).
- 
- **Desventajas**
    - ✓ La contratación de un plan de este software, implica una serie de funciones adicionales que no satisfacen o dan solución a la problemática presentada por el cliente, que llevo a la creación de SoftCook Management.



- ✓ Se centra en el área de ventas, tanto en los pedidos como en el control de caja.

#### 4.6.2. Comparativo entre las funciones principales de las 3 aplicaciones de Gestión de Restaurant

Funciones	SoftCook	Resto	SmartRestoBar
Administración de caja		✓	✓
Gestión de mesas		✓	✓
Administración de inventario	✓	✓	✓
Emisión de boletas y facturas		✓	✓
Informes y estadísticas		✓	✓

Tipo de contrato Software	Contrato licencia completa	Contrato licencia anual (sin información de costos asociados)	Contrato licencia mensual y anual según plan adquirido standard, Premium y Smart
Tipo de plataforma de aplicación	Escritorio	Escritorio y Web	Web
Tipo de soporte	Soporte personalizado en la instalación y producción del software	Soporte telefónico y chat en línea	Soporte remoto, telefónico y chat en línea

Tabla 18: Cuadro comparativo de funcionalidades

Dentro de las posibilidades de contratación del servicio de las aplicaciones que hemos comparado, estas aplicaciones poseen un contrato anual sobre los componentes que el cliente desea contratar.

Si bien, el sistema de SoftCook Management consta de elementos que permiten que sea completo basándose en la solicitud del cliente, es importante destacar que dentro del mercado existen alternativas de competencia para este sistema.

A continuación, se detallarán dos aplicaciones encontradas en sitios web, de los cuales detallaremos.

### **Alternativa 1:**

Para la plataforma **SmartRestoBar** posee costos por implementación y planes con módulos implementados.

- **Plan Standard**

Implementación: 6 UF
----------------------

Costo Mensual: \$55.000.-
Costo Anual: \$560.000.-

*Tabla 19: Costos del Plan Standard - SmartRestoBar*

Módulos implementados:

- ✓ Hasta 10 mesas
- ✓ Control Caja
- ✓ Control de Stock inventario
- ✓ Actualizaciones Gratuitas
- ✓ Acceso APP Android y Windows

- **Plan Premium**

Implementación: 13 UF
Costo Mensual: \$75.000.-
Costo Anual: \$760.000.-

*Tabla 20: Costos del Plan Premium - SmartRestoBar*

Módulos implementados:

- ✓ Hasta 30 mesas
- ✓ Control Caja
- ✓ Punto de venta garzón
- ✓ Cocina Touch
- ✓ Barra Touch
- ✓ Control de Stock inventario
- ✓ Actualizaciones Gratuitas
- ✓ Acceso App Android y Windows
- ✓ Soporte Continuo

- **Plan Smart**

Implementación: 16 UF
Costo Mensual: \$95.000.-
Costo Anual: \$960.000.-

*Tabla 21: Costos del Plan Smart - SmartRestoBar*

Módulos Implementados:

- ✓ Sin límites de mesas
- ✓ Control Caja
- ✓ Punto venta garzón
- ✓ Cocina Touch
- ✓ Barra Touch
- ✓ Control de Stock inventario
- ✓ Reportes Automáticos
- ✓ Actualizaciones Gratuitas
- ✓ Acceso App Android y Windows
- ✓ Soporte Continuo

Observación: Para obtener los valores de la aplicación SmartRestoBar, estos fueron copiados desde su página web <http://smartrestobar.cl/>.

## **Alternativa 2:**

Para la plataforma **Resto**, posee distintos módulos a contratar como Pc de ventas e inventarios y de compras donde también se incluye sistema de facturación electrónica.

### **Costos:**

Licenciamiento e implementación de sistema Resto 3 Pc \$1.294.560 neto de IVA
Up-Grade 18 UF + IVA disponibilidad Anual de nueva versión de Resto

*Tabla 22 : Costos Plataforma Resto*

Dentro de la cotización realizada por Axoft Chile S.A Empresa creadora de Resto, señala que la adquisición del software es totalmente física y que solo los informes (Nexo Reportes) y respaldos (Nexo Backup) son realizados mediante la nube.

Ya adquirido el software Axoft Chile S.A pone a disposición del cliente 30 horas de asesoría para la implementación presencial donde incluyen habilitación de boletas, facturas, guía de despacho y notas de crédito electrónica.

En el plan de Up-Grade del software Axoft Chile S.A dispondrá una actualización de versión cada año. El cliente deberá anualmente de contratar la actualización (18 UF + IVA = \$615.230.- IVA Incl.) así podrá obtener soporte sobre el software, así como la operación de los sistemas de acceso a Tablet, facturación, guías de despachos, notas de crédito electrónica y nexos reportes.

Para el servicio de soporte del sistema Axoft Chile S.A entrega soporte telefónico y remoto los 365 días del año y mediante una línea telefónica fija los cuales tendrán un horario de 9:30 a 18:30 y un número celular de turno desde las 18:30 a 00:00. Para los días sábado, domingo y festivos el soporte es desde las 10:00 a 00:00.

Módulos implementados:

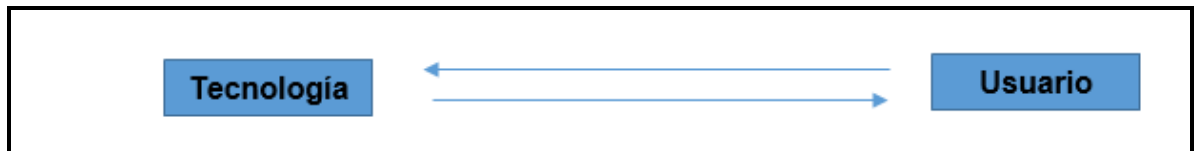
- ✓ Venta Salón
- ✓ Caja
- ✓ Venta Fast Food

- ✓ Cocina
- ✓ Delivery
- ✓ Inventario
- ✓ Nexo Mobile
- ✓ Nexo Reportes
- ✓ Nexo Notificaciones
- ✓ Nexo Update

Observación: Para obtener los valores de la aplicación Resto, se contactó vía mail al Product Manager Sr. Cristian González Cea ([cristiang@axoft.cl](mailto:cristiang@axoft.cl)) y él nos proporcionó la información de los costos reales de esta implementación.

#### 4.6.3. Uso de Tecnologías

Independientemente de cuál sea la tecnología implementada siempre es necesario saber que el cliente podrá, de algún modo, entenderse con la tecnología planteada, lo cual haría más eficiente incorporar nuevas alternativas en el mercado.



*Figura 35: Uso de Tecnologías*

- **Alternativas:**

Considerando el entorno en el cual va a ejecutar **SoftCook Management** y el mercado el cual va a abarcar este sistema, es que se originan las alternativas en el mercado.

#### 4.6.4. Software Propio

Un software propio, es del tipo de software el cual nace dentro de la empresa en la que se va a ejecutar, adaptando las características necesarias de esta y optando por los funcionamientos precisos para la eficacia del uso del sistema. Según la dependencia económica de la empresa es el equipo de desarrollo que podrá llevar a cabo la construcción de la aplicación.

Existen alternativas distintas para todas las empresas según su categoría:

**Grandes empresas:** Cuentan con un departamento enfocado a cada área profesional, será posible integrar el sistema con desarrolladores internos de la empresa, los cuales serán supervisados por los ingenieros creadores de SoftCook Management, pero toda estructura será creada por ellos.

**Medianas empresas:** Es posible trabajar con desarrolladores externos, en este caso, desarrolladores de SoftCook Management. Los cuáles serán los arquitectos e ingenieros del sistema por completo.

- **Ventajas**

- ✓ Posee lo necesario
- ✓ Se Integra con otros softwares
- ✓ No depende de terceros (excepto supervisores de SoftCook Management).

- **Desventajas**

- ✓ Es necesario tener un equipo desarrollador y técnico interno o externo, lo cual conllevaría a un gasto económico más.
- ✓ Alto costo
- ✓ Solicitud en el reglamento gubernamental
- ✓ Fragilidad de soluciones

#### 4.6.5. Desarrollo Externo

El desarrollo externo para una empresa desarrolladora suele ser más factible, ya que la misma empresa suele hacerse cargo del desarrollo, en este caso desarrolladores directos de SoftCook Management, codificación, actualizaciones y licencias que son necesarias para el funcionamiento correcto del sistema, las cuales pueden abarcar varias empresas simultáneamente, y con esto ahorrar costos y tiempo.

- **Ventajas**
  - ✓ Licenciar el software desarrollado
  - ✓ Operaciones a través Housing, Hosting, etc.
  - ✓ Explotación a través de Outsourcing
- **Desventajas**
  - ✓ Leve dificultad en el traspaso de información

#### **4.6.6. Operación de Sistema Externo**

La operación de sistema externo es de un punto de vista conveniente para el desarrollador, ya que este se ejecuta a través de internet, lo cual disminuye, en cierto modo, el asesoramiento de ayuda técnica presencial.

- **Ventajas**
  - ✓ Disminuye los expertos informáticos.
  - ✓ Se necesita un hardware de funcionamiento básico.
  - ✓ Acceso al sistema a través de la web.
- **Desventajas**
  - ✓ Dependencia de proveedor externo
  - ✓ Dependencia de Internet
  - ✓ Mayor tiempo de respuesta

#### **4.6.7. Explotación de Sistemas Externos BPO**

BPO (Business Process Outsourcing o subcontratación de procesos de negocios) es la subcontratación de funciones del proceso de negocios en



proveedores de servicios, ya sea internos o externos a la empresa, que se suponen menos costosos o más eficientes y eficaces. Básicamente BPO sugiere a sus clientes “confiar en ellos”, lo que permite disminuir preocupaciones en cuanto al desarrollo y mantenimiento del software.

- **Ventajas**

- ✓ La empresa desarrolladora se encarga de entregar la responsabilidad a BPO sobre la contratación del sistema.
- ✓ Dado a que un BPO mantiene varias empresas expertas a su cargo, es posible disminuir la solicitud de servicios técnicos.
- ✓ Acceso de manera mundial y nacional.

- **Desventajas**

- ✓ Alto costo
- ✓ Las empresas deben modificar los procesos internos
- ✓ Sobrevivencia del sistema en manejo de agentes externos

#### 4.6.8. Desarrollo de software

- ✓ Definición de requisitos: Documentos de especificación.
- ✓ Diseño funcional del sistema: Diseño del flujo funcional del programa.
- ✓ Diseño técnico del sistema: Definición de arquitectura e interfaces.
- ✓ Especificación de los componentes: Estructura de los componentes.
- ✓ Programación: Creación de código ejecutable.

#### 4.7. ¿Qué aspectos debe contemplar una prueba bien definida?

PROPIEDAD	SIGNIFICADO
✓ Identificador	Código único de la prueba

✓ Valores de entrada	Descripción de los datos de entrada de un objeto de prueba
✓ Resultados esperados	Datos de salida que se espera se produzcan
✓ Precondiciones	Situación previa a la ejecución de la prueba o características de un objeto de prueba antes de ejecutar un caso de prueba
✓ Postcondiciones	Características de un objeto de prueba tras la ejecución de la prueba
✓ Dependencias	Relación u orden de dependencia entre casos de pruebas
✓ Acciones	Pasos que llevar a cabo para ejecutar la prueba
✓ Requisito vinculado	Relación de requisitos que se pretenden validar con la ejecución de la prueba

*Tabla 23: Aspectos de pruebas bien definidas*

#### **4.8. ¿Que son las pruebas automatizadas?**

Las pruebas automatizadas multiplican la capacidad de los equipos de testing o unidades de Calidad de las organizaciones. Minimizan sustancialmente la ejecución de las pruebas manuales, requiriéndose estas últimas solo en escenarios especiales.

De este modo el recurso humano puede realizar otro tipo de tareas, mientras la automatización se ocupa de realizar las pruebas relevantes y repetitivas.

Por otra parte, permite ejecutar mayor cantidad de pruebas, sin necesidad de incrementar el número de personas dedicado a testear, esto hace que el equipo de trabajo sea mucho más eficiente y productivo, haciendo rentable la ejecución de pruebas de software.

Las pruebas automatizadas facilitan la precisión a la hora de diagnosticar la falla detectada y proveen la evidencia en una serie de reportes, así como un video que permiten analizar en profundidad la situación. Al utilizar distintos juegos de datos en el mismo caso de prueba, es posible ser muy certero en el diagnóstico de

las situaciones problemáticas. El testing automatizado también permite mayor cubrimiento de código y de casos, por lo que se logran pruebas muy exhaustivas y precisas de la funcionalidad, y más condiciones probadas.

Para alcanzar este objetivo, el uso de metodologías ágiles y automatización de ciertos procesos suelen ser estrategias clave.

El uso de metodologías ágiles se basa en la comprobación continua del código para poder integrar poco a poco al código ya existente, las nuevas funcionalidades o mejoras, actualizando y verificando a diario para obtener un resultado mucho más fiable y en el menor tiempo posible. Esto implica que en cada ciclo de desarrollo se ejecuten los mismos casos de prueba con la finalidad de asegurar que las modificaciones que se hicieron en el código no generen defectos.

La automatización se encarga de agilizar cualquier proceso reduciendo el tiempo en el que se prueba el producto.

#### **4.8.1. Ventajas Pruebas Automatizadas**

- ✓ **Fiabilidad:** Se elimina el error humano. El script tendrá pasos precisos que serán ejecutados siempre de la misma manera. Ejecutar la misma tarea repetitivamente puede volverse tedioso y después de cierto tiempo el tester puede no prestar la misma atención a los detalles de dicha prueba y omitir algún tipo de error. Al ejecutarse repetidamente como parte del desarrollo permite a los desarrolladores comprobar que sus modificaciones no introdujeron errores en partes críticas de la aplicación.
- ✓ **Rapidez:** Por ser una prueba de naturaleza automática, será ejecutada mucho más rápido que si la ejecutara un tester. Esto dependerá mucho de la infraestructura con la que se cuente y el nivel de integración de las pruebas. No es lo mismo que las pruebas automáticas se lancen de forma automática cada vez que se deploya la aplicación y corran en un ambiente dedicado, a que, como ocurre en muchos casos, deba iniciarlas manualmente un tester, en su computadora, para que corran en un

ambiente compartido (como QA<sup>10</sup> o UAT<sup>11</sup>) donde alguien puede cambiar un dato y la prueba podría fallar.

#### **4.8.2. Desventajas Pruebas Automatizadas**

- ✓ La creación de scripts automatizados es un proceso costoso que requiere un gran nivel de habilidades técnicas.
- ✓ No se pueden automatizar una aplicación al 100%. por lo general, las aplicaciones siempre van a tener alguna funcionalidad que requiera la interacción del usuario, por ejemplo, hacer clic en un link que se envió en un mail de confirmación (este caso podría llegar a automatizarse en algunos casos), o comparar que una imagen sea exactamente igual a la esperada (cada browser la mostrará levemente diferente) o un login que solicita seleccionar imágenes para comprobar que no se trata de un robot, se requieren de intervención del tester y no puede ser automatizado.
- ✓ Siempre hay que tener en cuenta cual es el costo y cuál es el beneficio, para evaluar si las pruebas se van a realizar de forma manual o automatizada, teniendo en cuenta el conocimiento de la herramienta a utilizar, lenguajes de programación, tiempos y demás.

#### **4.9. Pruebas Unitarias e Integración**

Las pruebas de integración son imprescindibles, ya que nos permiten garantizar que las aplicaciones cumplen las funcionalidades que se esperan de ellas y las expectativas de calidad (no solo de código); ayudando a encontrar esos errores o defectos que aún no se han descubierto; reduciendo el costo del desarrollo, el de propiedad para los usuarios; y desarrollar confianza en los clientes al evitar los molestos errores de regresión.

Las pruebas Unitarias se ejecutan en la primera etapa donde el desarrollador ya cuenta con los componentes del sistema desarrollado, el

---

<sup>10</sup> QA : Quality Assurance – Aseguramiento de Calidad.

<sup>11</sup> UAT: User Acceptance Testing – Prueba de aceptación por parte de los usuarios.

desarrollador se encarga que todos sus componentes funcionen de acuerdo con lo esperado.

Por otro lado, las pruebas de integración se ejecutan una vez concluidas las pruebas de componentes o unitarias, estas se ejecutan cuando se tiene la mayor parte de componentes integrados y así verificar que todos operen correctamente

Para la realización de un test unitario o de componentes debemos ingresar en NetBeans y revisar que tenemos instalado JUnit el cual nos ayudará a realizar las pruebas Unitarias y de integración. Para eso debemos ingresar al componente que vamos a realizarle pruebas e ir a Tool / Create/Update Test.

#### **4.9.1. Pruebas Unitarias - Alcance**

- ✓ Solo se prueban componentes individuales (clases, funciones, módulos).
- ✓ Cada componente se prueba de forma independiente, en donde se logra descubrir errores producidos por defectos internos.
- ✓ Los casos de prueba podrán ser obtenidos a partir de código fuente, modelo de datos, diseño de software.
- ✓ Se pueden realizar pruebas de caja negra para probar los módulos completamente.

#### **4.9.2. Pruebas de integración – Alcance**

- ✓ Las pruebas de integración comprueban la interacción mutua entre los componentes entre sí.
- ✓ Integración no incremental: combinar los módulos y probar el sistema en su conjunto.
- ✓ El software se prueba en pequeñas porciones, en la detección y resolución de errores es más fácil y controlable.
- ✓ Los casos de prueba podrán ser obtenidos a partir de la especificación de interfaces, diseño de la arquitectura y el modelo de datos.

#### 4.10. Pruebas automatizadas con JUnit

Se realizó las pruebas unitarias a las App\_Cocina y App\_Stock donde se revisó con la base de datos el listado de ingredientes.

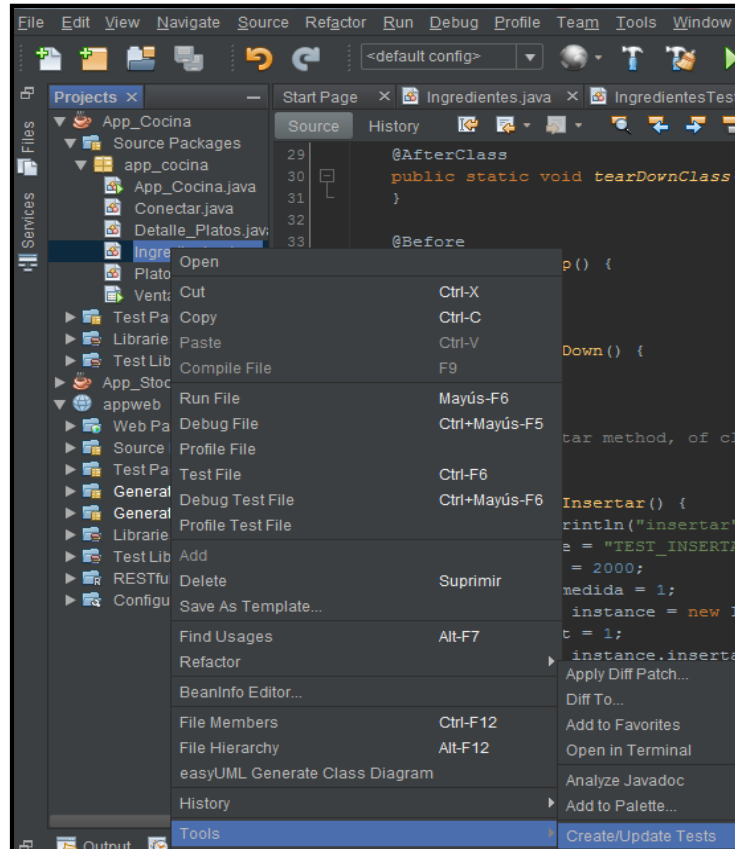


Figura 36: Prueba automatizada

Acá se nos abrirá una ventana donde nos aparecerá las opciones para la realización de la prueba y que tipo de frameworks, vamos a utilizar por nuestra parte JUnit.

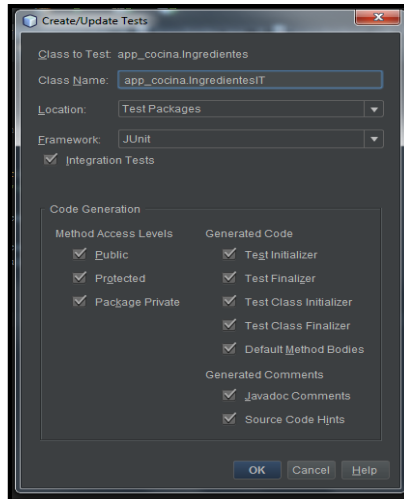


Figura 37: Realización de Prueba

Ya creado el archivo de prueba, dentro de las pruebas se llamará a las librerías necesarias para realizarlas (complemento de JUnit).

```
package app_cocina;

import java.sql.ResultSet;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;
```

Figura 38: Librerías

Y de forma automatizada realizará los métodos para la realización de Pruebas.

Test de insertar Ingrediente:

```

// @Test
public void testInsertar() {
    System.out.println("insertar");
    String nombre = "TEST";
    int cantidad = 1000;
    int id_tipo_medida = 1;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.insertar(nombre, cantidad, id_tipo_medida);
    assertEquals(expectedResult, result);
}

```

Figura 39: Test Insertar Ingrediente

Test de Actualización de Ingredientes:

```

/**
 * Test of actualizar method, of class Ingredientes.
 */
@Test
public void testActualizar() {
    System.out.println("actualizar");
    int id = 42;
    String nombre = "TEST";
    int cantidad = 500;
    int id_tipo_medida = 1;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.actualizar(id, nombre, cantidad, id_tipo_medida);
    assertEquals(expectedResult, result);
}

```

Figura 40: Test Actualización Ingredientes

Test de eliminación de Ingredientes:

```

@Test
public void testEliminar() {
    System.out.println("eliminar");
    int id = 42;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.eliminar(id);
    assertEquals(expectedResult, result);
}

```

Figura 41: Test Eliminación Ingredientes

Realización de test:



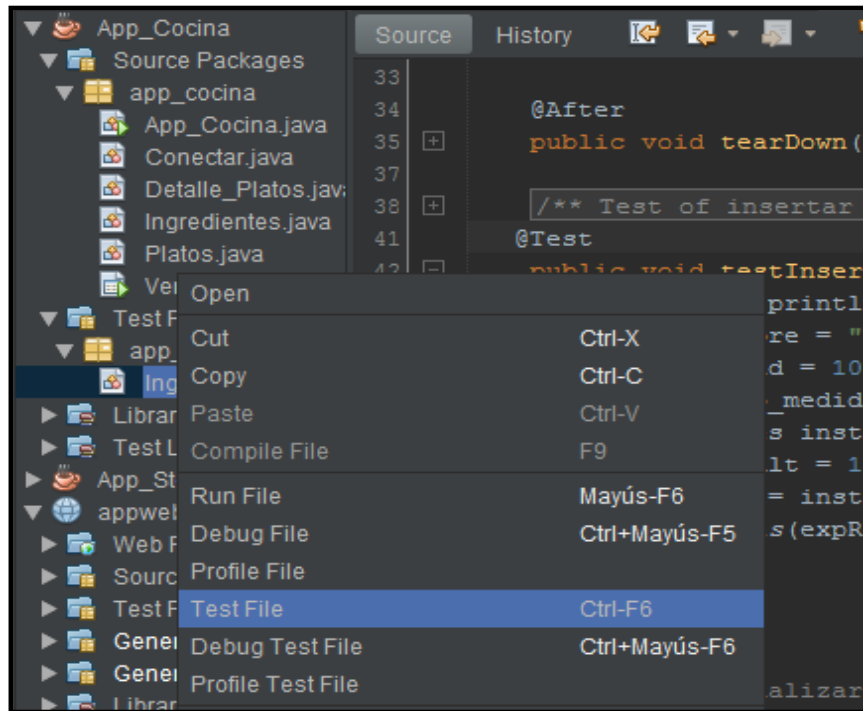


Figura 42: Realización Test

Se revisa en la Base de datos que no tenemos alguna Prueba realizada dentro de la tabla ingredientes.

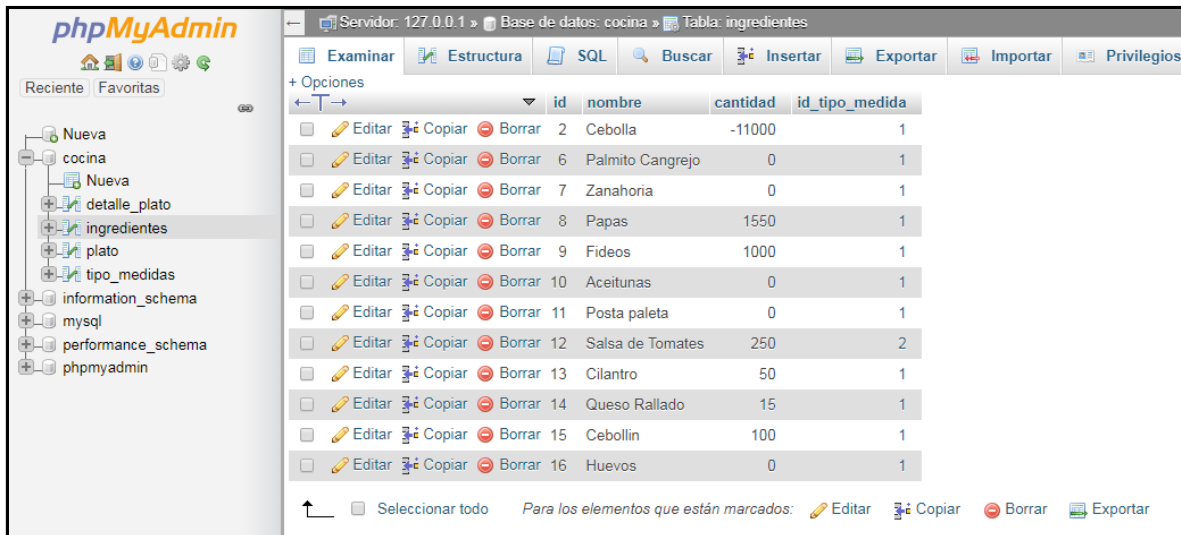


Figura 43: Revisión Pruebas

Realizamos la prueba de Inserción de un Ingrediente con su medida y cantidad.

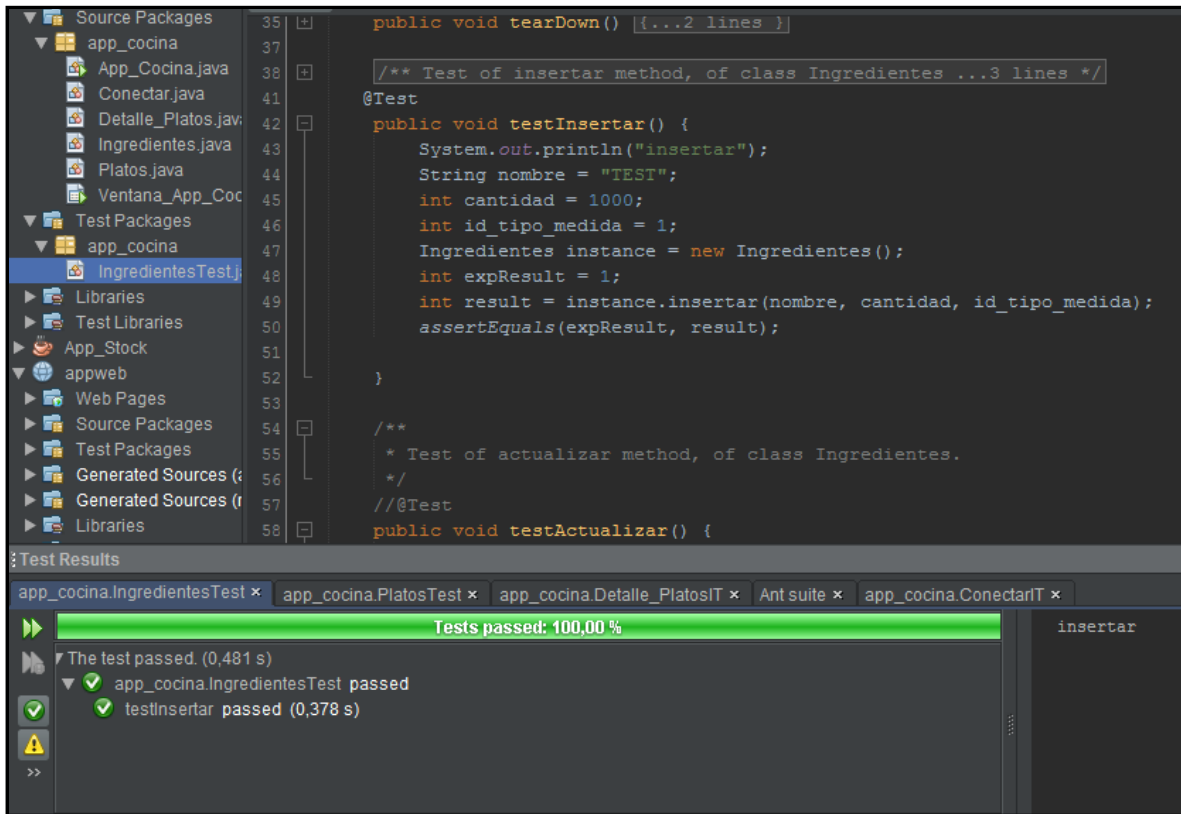


Figura 44: Prueba de Inserción

Revisamos la base de datos MySQL si fue insertado el Test con su cantidad y tipo de medida.

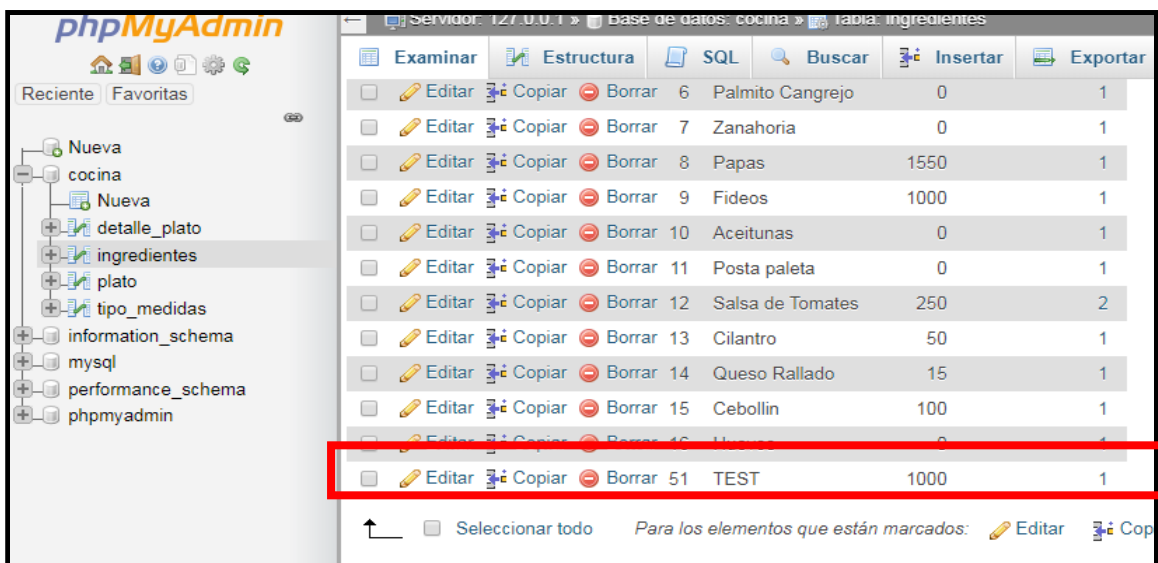


Figura 45: Revisión de Base Datos

Se realiza el test de actualización de los ingredientes

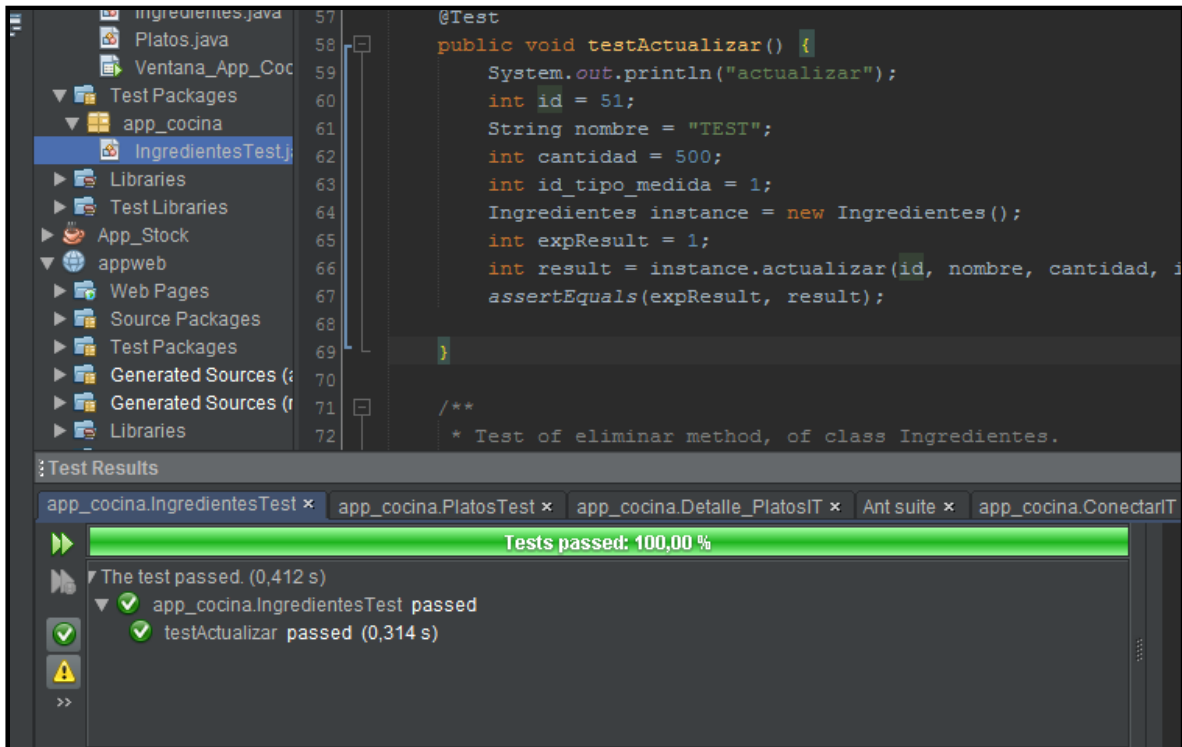


Figura 46: Realización Test Actualización

Revisamos la Base de datos si la modificación fue realizada:

The screenshot shows a database management tool interface. On the left, there is a tree view of the database structure. On the right, a table of ingredients is displayed. The table has the following columns: id, nombre, cantidad, and id\_tipo\_medida. The data is as follows:

id	nombre	cantidad	id_tipo_medida
2	Cebolla	-11000	1
6	Palmito Cangrejo	0	1
7	Zanahoria	0	1
8	Papas	1550	1
9	Fideos	1000	1
10	Aceitunas	0	1
11	Posta paleta	0	1
12	Salsa de Tomates	250	2
13	Cilantro	50	1
14	Queso Rallado	15	1
15	Cebollin	100	1
16	Huevos	0	1
51	TEST	500	1

Figura 47: Revisión Base de Datos

Y por último se realiza el test de eliminación en la base de datos.

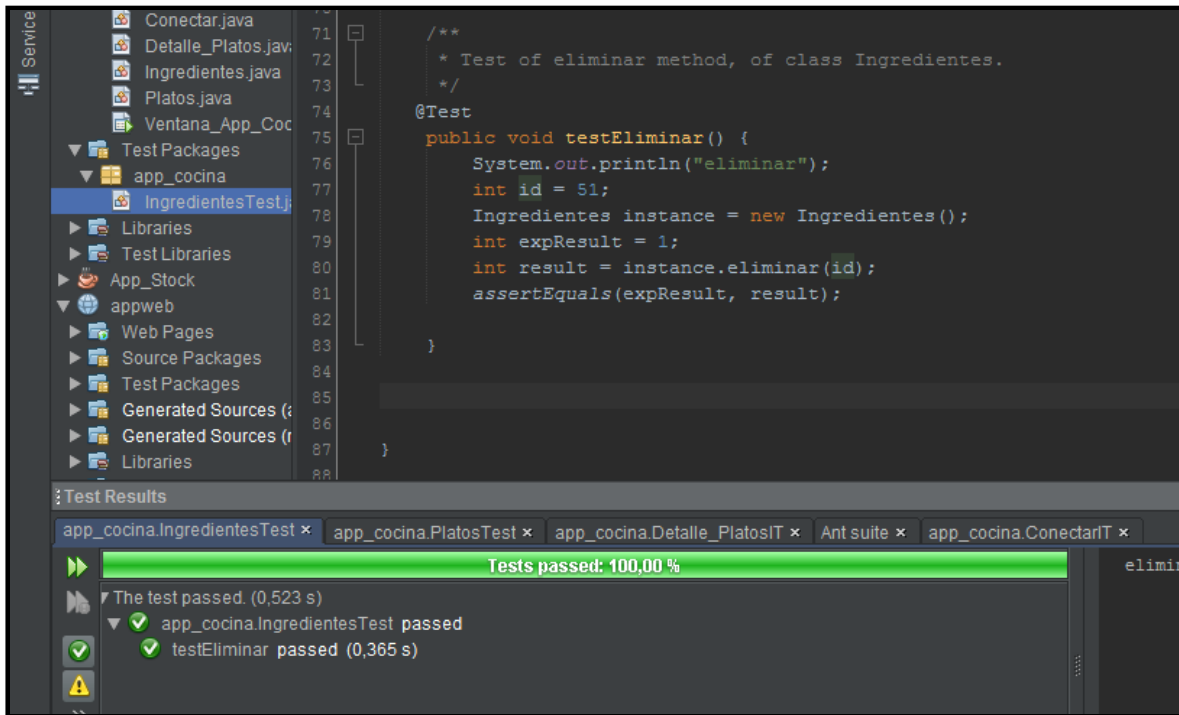


Figura 48: Realización Test Eliminación

Revisamos si el test fue eliminado de la base de datos y se ratifica que no está ya disponible el ingrediente.

The screenshot shows a database management tool interface. On the left is a tree view with 'cocina' expanded to show 'detalle\_plato', 'ingredientes', 'plato', and 'tipo\_medidas'. The main area displays a table of ingredients:

ID	Nombre	Cantidad	Estado
6	Paimito Cangrejo	0	1
7	Zanahoria	0	1
8	Papas	1550	1
9	Fideos	1000	1
10	Aceitunas	0	1
11	Posta paleta	0	1
12	Salsa de Tomates	250	2
13	Cilantro	50	1
14	Queso Rallado	15	1
15	Cebollin	100	1
16	Huevos	0	1

Figura 49: Revisión Test Eliminado

Se realizó las pruebas unitarias a las App\_Cocina y App\_Stock donde se revisó con la base de datos el listado de ingredientes.

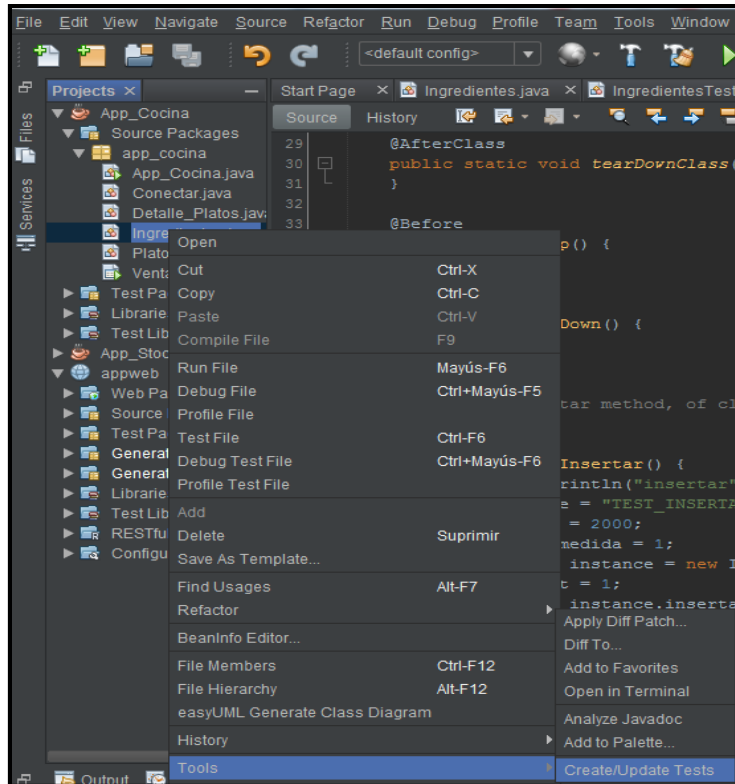


Figura 50: Pruebas Unitarias App\_Cocina y App\_Stock

Acá se nos abrirá una ventana donde nos aparecerá las opciones para la realización del test y que tipo de frameworks, vamos a utilizar por nuestra parte JUnit.

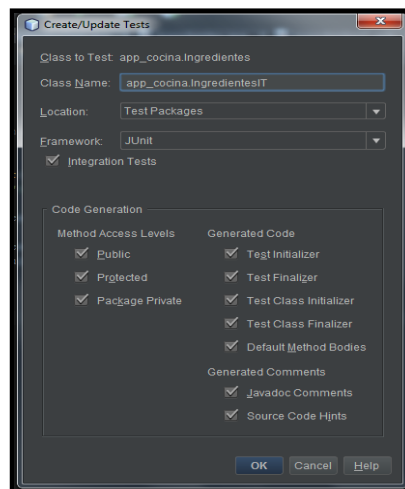


Figura 51: Opciones para realizar test

Ya creado el archivo de prueba, dentro de las pruebas se llamará a las librerías necesarias para realizarlas (complemento de JUnit).

```

package app_cocina;

import java.sql.ResultSet;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

```

Figura 52: Llamado Librerías

Y de forma automatizada realizará los métodos para la realización de Pruebas.

Test de insertar Ingrediente:

```

// @Test
public void testInsertar() {
    System.out.println("insertar");
    String nombre = "TEST";
    int cantidad = 1000;
    int id_tipo_medida = 1;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.insertar(nombre, cantidad, id_tipo_medida);
    assertEquals(expectedResult, result);
}

```

Figura 53: Métodos – Test Insertar

Test de Actualización de Ingredientes:

```

/**
 * Test of actualizar method, of class Ingredientes.
 */
@Test
public void testActualizar() {
    System.out.println("actualizar");
    int id = 42;
    String nombre = "TEST";
    int cantidad = 500;
    int id_tipo_medida = 1;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.actualizar(id, nombre, cantidad, id_tipo_medida);
    assertEquals(expectedResult, result);
}

```

Figura 54: Métodos – Test Actualizar

Test de eliminación de Ingrediente:

```
@Test
public void testEliminar() {
    System.out.println("eliminar");
    int id = 42;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.eliminar(id);
    assertEquals(expectedResult, result);
}
```

Figura 55: Test Eliminación

Realización de test:

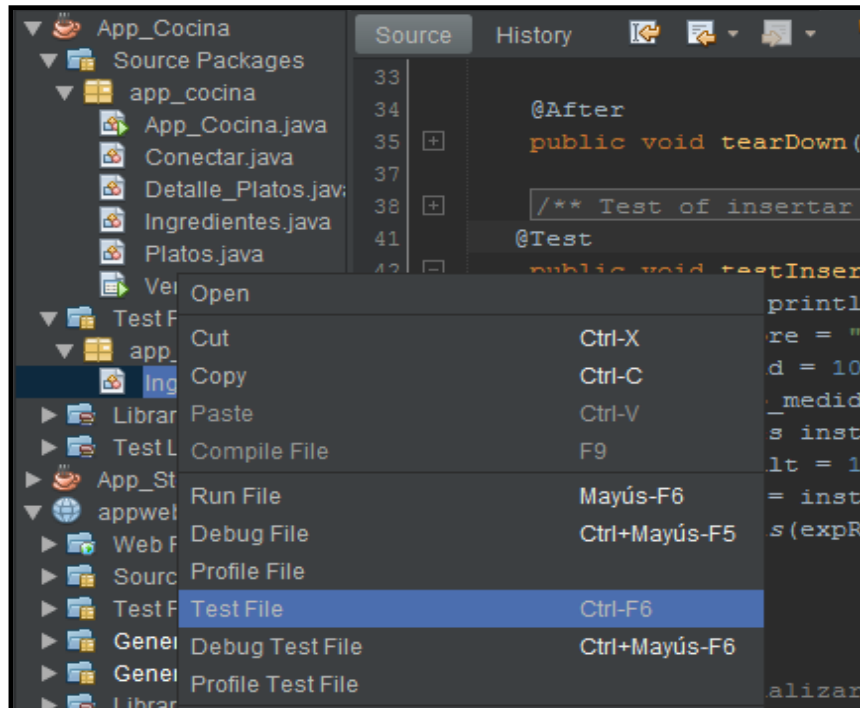


Figura 56: Realización Test

Revisamos en la Base de datos que no tenemos alguna Prueba realizada dentro de la tabla ingredientes.



Figura 57: Revisión Base Datos

Realizamos la prueba de Inserción de un Ingrediente con su medida y cantidad.

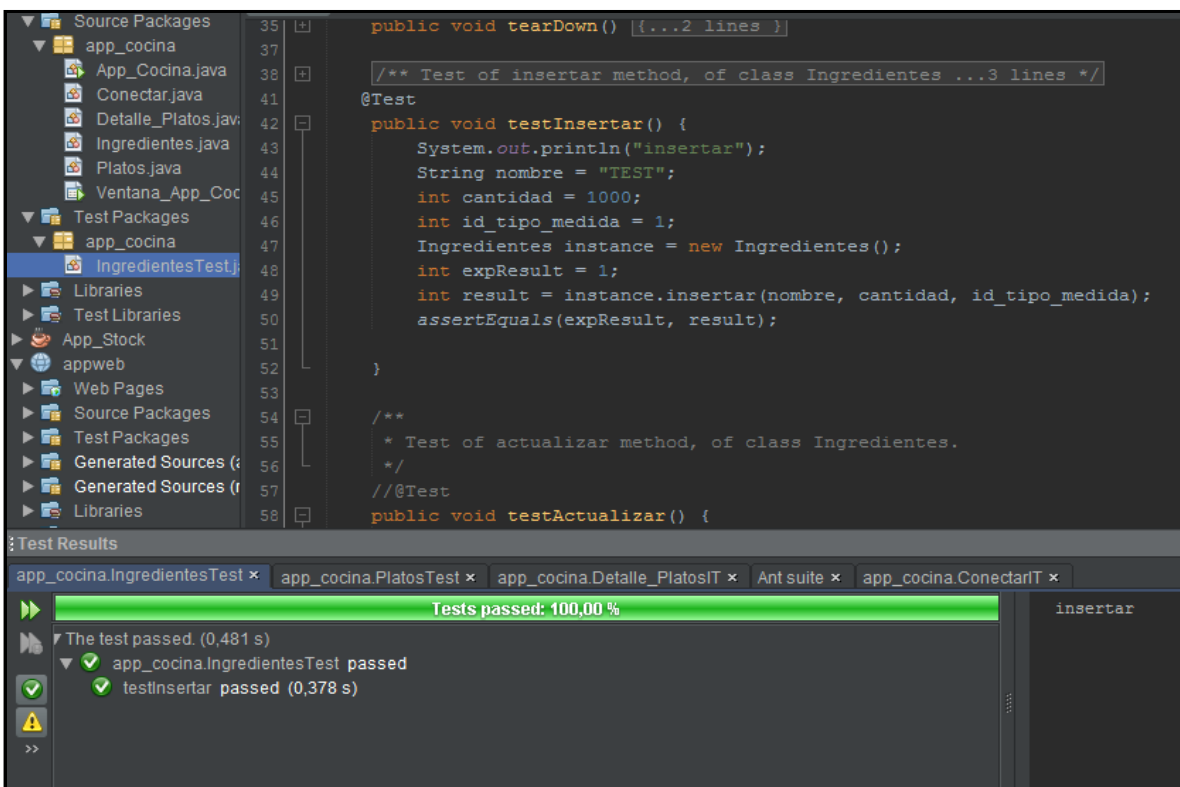


Figura 58: Realización Inserción

Revisamos la base de datos MySQL si fue insertado el Test con su cantidad y tipo de medida.



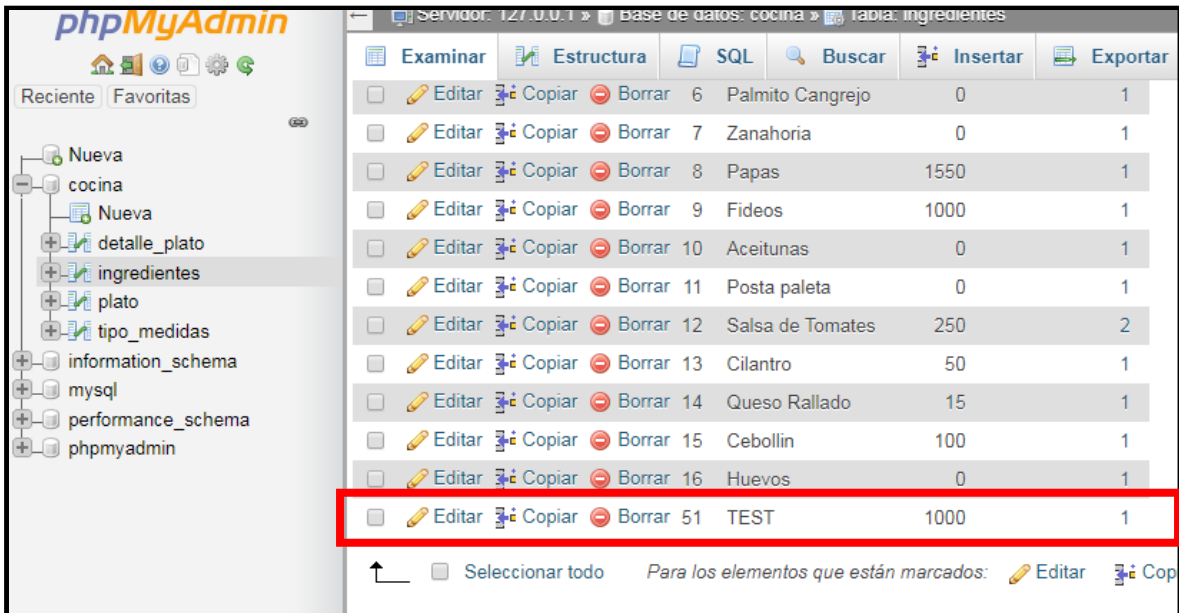


Figura 59: Revisión Base Datos en phpMyAdmin

Se realiza el test de actualización de los ingredientes

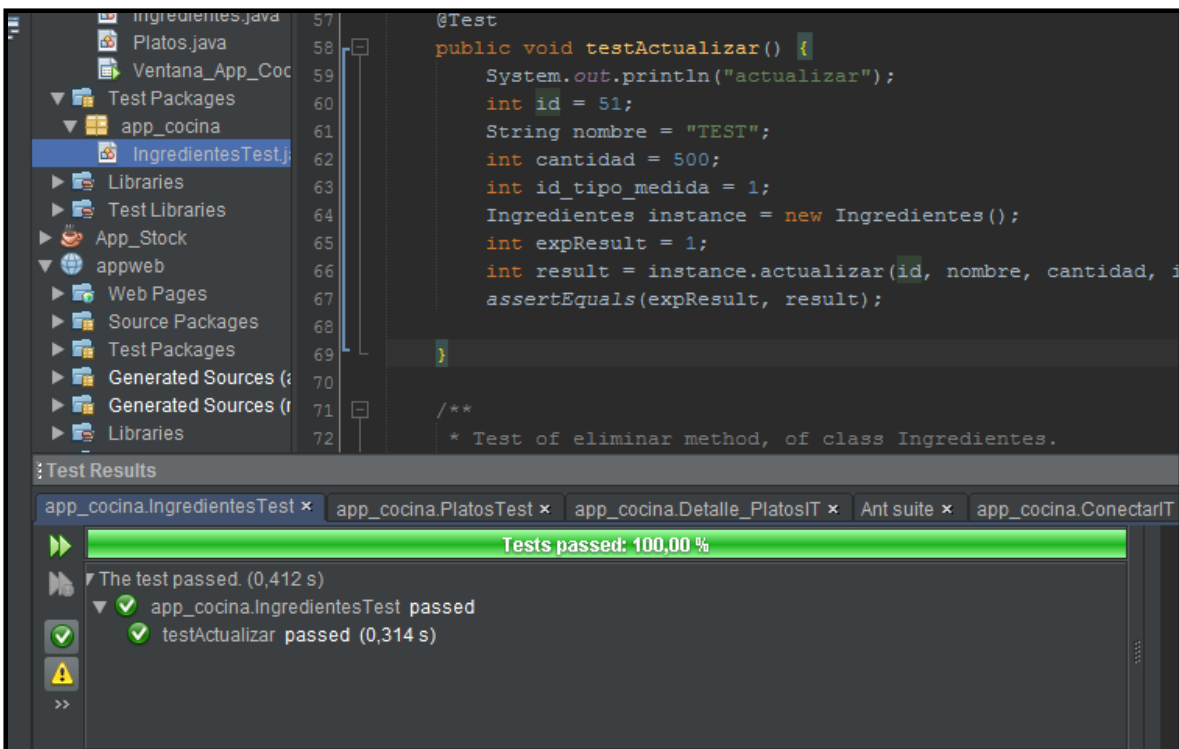


Figura 60: Test Actualización Realizado

Revisamos la Base de datos si la modificación fue realizada:

	id	nombre	cantidad	id_tipo_medida
<input type="checkbox"/>	2	Cebolla	-11000	1
<input type="checkbox"/>	6	Palmito Cangrejo	0	1
<input type="checkbox"/>	7	Zanahoria	0	1
<input type="checkbox"/>	8	Papas	1550	1
<input type="checkbox"/>	9	Fideos	1000	1
<input type="checkbox"/>	10	Aceitunas	0	1
<input type="checkbox"/>	11	Posta paleta	0	1
<input type="checkbox"/>	12	Salsa de Tomates	250	2
<input type="checkbox"/>	13	Cilantro	50	1
<input type="checkbox"/>	14	Queso Rallado	15	1
<input type="checkbox"/>	15	Cebollin	100	1
<input type="checkbox"/>	16	Huevos	0	1
<input type="checkbox"/>	51	TEST	500	1

Figura 61: Revisión en Base de Datos

Y por último se realiza el test de eliminación en la base de datos.

```

71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
/**
 * Test of eliminar method, of class Ingredientes.
 */
@Test
public void testEliminar() {
    System.out.println("eliminar");
    int id = 51;
    Ingredientes instance = new Ingredientes();
    int expectedResult = 1;
    int result = instance.eliminar(id);
    assertEquals(expectedResult, result);
}

```

Test Results

app\_cocina.IngredientesTest × app\_cocina.PlatosTest × app\_cocina.Detalle\_PlatosIT × Ant suite × app\_cocina.ConectarIT ×

Tests passed: 100,00 %

The test passed. (0,523 s)

- app\_cocina.IngredientesTest passed
  - testEliminar passed (0,365 s)

Figura 62: Realización Test Eliminación

Revisamos si el test fue eliminado de la base de datos y se ratifica que no está ya disponible el ingrediente.

ID	Nombre	Cantidad	Unidad
6	Palmito Cangrejo	0	1
7	Zanahoria	0	1
8	Papas	1550	1
9	Fideos	1000	1
10	Aceitunas	0	1
11	Posta paleta	0	1
12	Salsa de Tomates	250	2
13	Cilantro	50	1
14	Queso Rallado	15	1
15	Cebollin	100	1
16	Huevos	0	1

Figura 63: Revisión Test Eliminado en phpMyAdmin

## 4.11. Pruebas de Sistema

Estas pruebas sirven para validar que todas las funciones y componentes del sistema colaboran correctamente, estas pruebas se ejecutan una vez concluidas las pruebas unitarias y las de integración.

### 4.11.1. Alcance

- ✓ Consiste en probar el software completo para verificar que se cumplen los requisitos funcionales establecidos, y se cumplen los aspectos no funcionales de calidad, como lo es la usabilidad, eficiencia y seguridad,
- ✓ La calidad del software es observada desde el punto de vista del usuario y en un entorno de pruebas coincidente con el entorno real.
- ✓ Los casos de prueba podrán ser obtenidos a partir de las especificaciones funcionales, casos de uso, procesos de negocio.
- ✓ Para la generación de casos de prueba se utilizan técnicas de caja negra.

- En la aplicación APP Cocina, se muestra que existen 5 platos diferentes a Preparar, si selecciono el primer plato *Pollo con Salsa de Queso*, mostrará los Ingredientes necesarios para preparar este plato

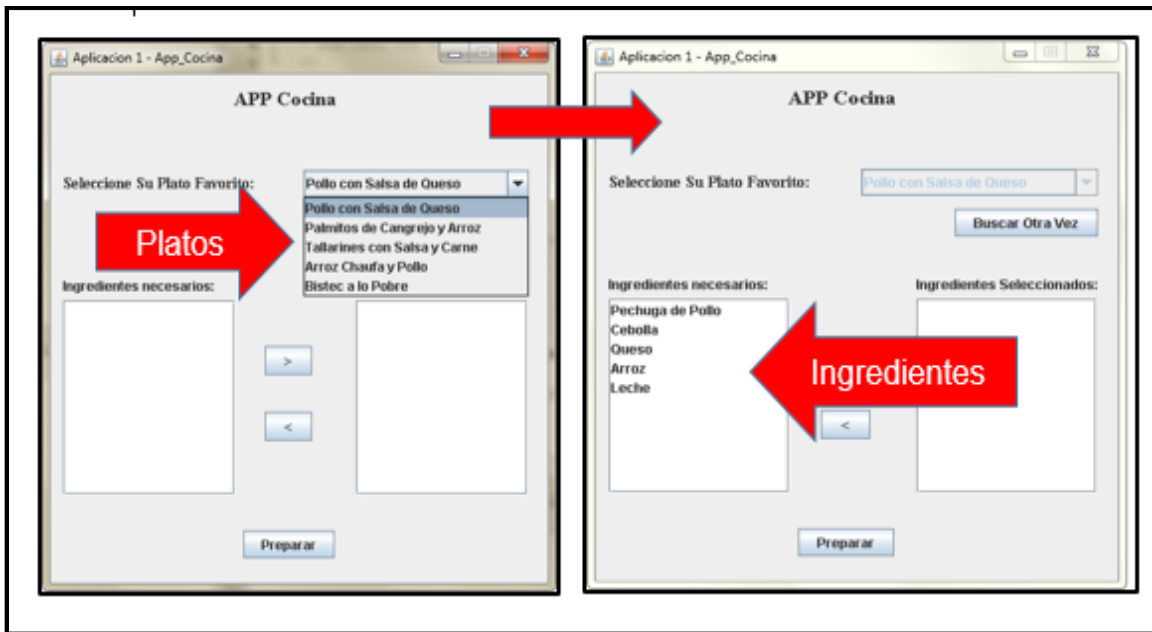


Figura 64: Muestra de platos a preparar

- Al seleccionar todos los Ingredientes hacia el costado derecho, la aplicación estará en condiciones de preparar el Plato elegido, al presionar el Botón Preparar, arrojará un mensaje de “El plato se ha preparado con éxito”.



Figura 65: Mensaje de plato preparado

- En la aplicación Stock, podemos ver los ingredientes de todos los posibles platos a elegir.

- Si me voy a la pestaña Editar, y al ingrediente Pechuga de Pollo le resto cantidad de gramos existente, por ejemplo, 400gramos, vuelvo a revisar las existencias, la cantidad de gramos se rebaja automáticamente a 100 gramos en existencia de bodega.

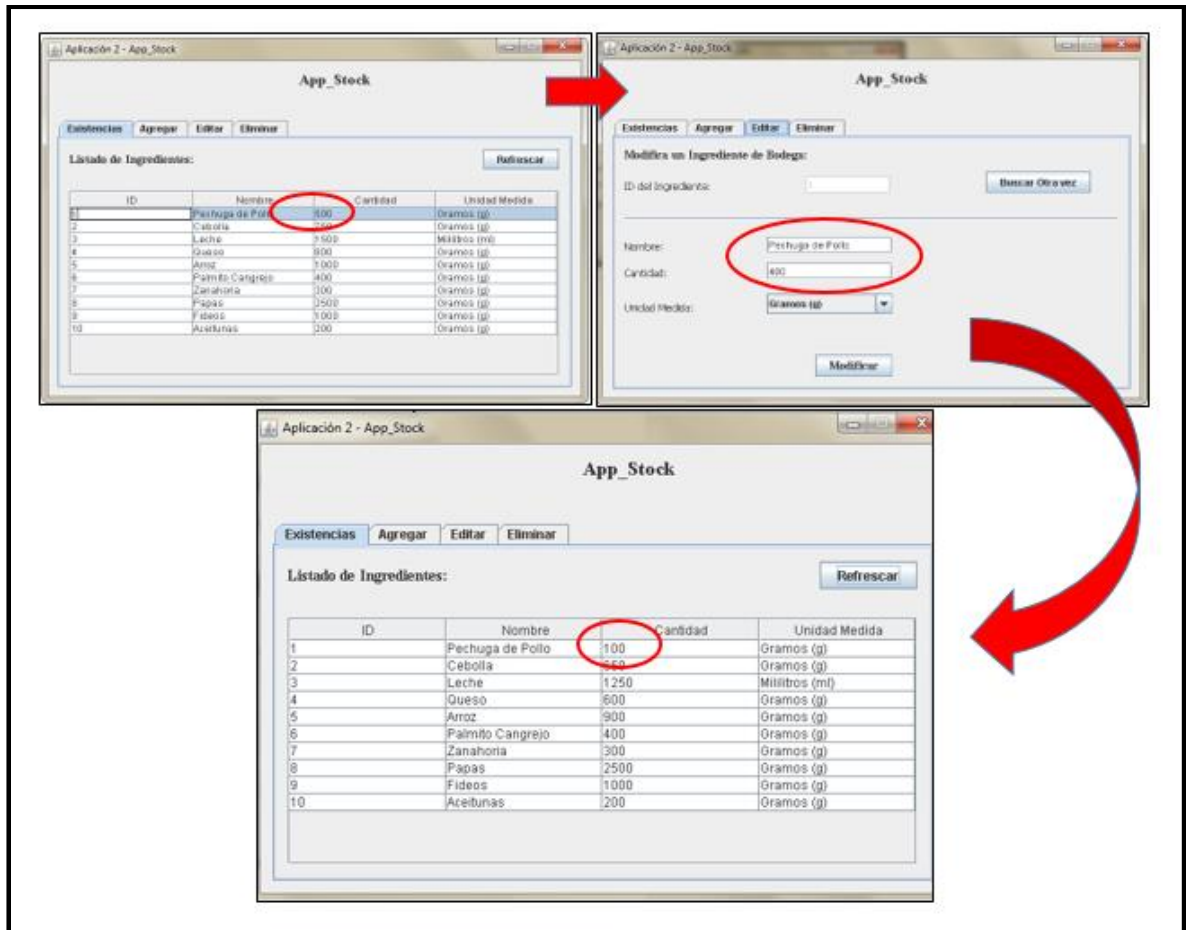


Figura 66: Revisión de existencias

- Revisión de la implementación.

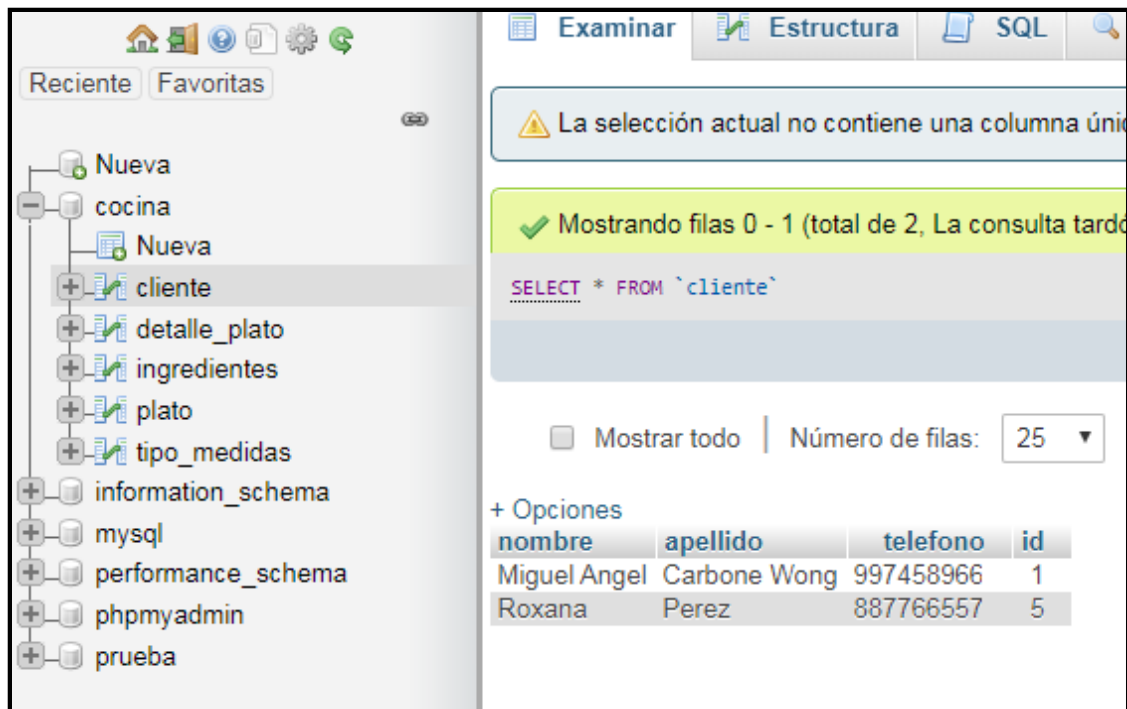


Figura 67: Revisión Implementación

#### 4.12. Pruebas de Caja Negra

Las pruebas de caja negra es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, se

enfoca básicamente en las entradas y salidas del sistema sin preocuparnos en tener el conocimiento de la estructura interna del programa de software.

#### App\_Cocina:

<b>Entradas Controladas</b>	<b>Salidas Esperadas</b>	<b>Salidas Obtenidas</b>	<b>Resultados</b>
Selección de platos	Menú de platos	Menú desplegable de platos	<b>Correcto</b>
Agregar ingredientes	Cambio de posición	Cambio de posición	<b>Correcto</b>
Agregar ingrediente	Mensaje de error	Mensaje	<b>Correcto</b>
Botón de preparar	Mensaje de éxito	Mensaje	<b>Correcto</b>
Actualización de BD	Actualización	Sin Mensaje	<b>Error</b>

Tabla 24: Prueba de Caja Negra App\_Cocina

#### App\_Stock:

<b>Entradas Controladas</b>	<b>Salidas Esperadas</b>	<b>Salidas Obtenidas</b>	<b>Resultados</b>
Actualización de Existencias	Cambio de ingredientes	Cambio de ingredientes	<b>Correcto</b>
Agregar Ingrediente	Mensaje de registro	Mensaje	<b>Correcto</b>
Edición de Ingredientes	Ingreso y Mensaje	Registro y Mensaje	<b>Correcto</b>
Eliminación de Ingredientes	Mensaje de Eliminación	Mensaje	<b>Correcto</b>

Tabla 25: Prueba de Caja Negra App\_Stock

Para las pruebas de caja negra se revisó el funcionamiento de las entradas y la obtención de las salidas correspondientes al sistema realizado.

#### 4.14. Mejora Continua para el proyecto SoftCook Management

Es necesario contemplar un proceso de mejora para nuestro Proyecto y sobre todo pensado en el restaurant Tayu, ya que los cambios tecnológicos y administrativos crecen en paralelo a la modernización, esta idea podría quedar

estipulada dentro de la factibilidad legal planteada con anterioridad, el objetivo de todo negocio es mantenerse, crecer, exitosos y por sobre todo rentables, la idea es permitir visualizar un horizonte amplio, en busca de la excelencia y la innovación.

El objetivo radica principalmente en mejorar los procesos y servicios que entregamos como equipo de desarrolladores, en procura de una estabilidad del proceso productivo que contempla las posibilidades de una o más mejoras.

- **Ventajas:**

- ✓ Mejorar las debilidades estipuladas en el análisis FODA realizado con anterioridad en este informe.
- ✓ Contribuye en la creación de una imagen más robusta y competitiva en el mercado.
- ✓ Facilita la corrección de errores detectados.
- ✓ Mejora la calidad del producto que ofrecemos, pensando en las necesidades de nuestros clientes.
- ✓ Minimiza las fallas en la calidad, con ello permite ahorrar dentro del dinero presupuestado.

- **Desventajas:**

- ✓ Puede ocasionar una inversión mayor a la hora de modernización del sistema.
- ✓ Requiere de un cambio en toda la organización, ya que el éxito del proyecto, dependerá de que todos los miembros del restaurant sean partícipes de las nuevas mejoras.
- ✓ Requiere de más tiempo, en la creación de las nuevas mejoras.
- ✓ Requiere de más riesgos que deben ser aceptados.

#### **4.14.1. Mejoras contempladas**

- ✓ Implementar una solución web que pueda visualizarse en distintas plataformas.



- ✓ Restringir el acceso al sistema, diseñando un usuario y contraseña a cada usuario del sistema.
- ✓ Implementar recuperación de clave a usuario del sistema.
- ✓ Implementar nueva funcionalidad, como, por ejemplo, informes de gestión.
- ✓ Agregar nueva funcionalidad al sistema, como, por ejemplo, contemplar el registro de los pedidos de los clientes, tal como muestra la imagen.

The screenshot shows a web interface for a restaurant. At the top, it says "Restaurant" and "Pedidos Clientes". Below this is a form with four input fields: "Nombre:", "Apellido:", "telefono:", and "Id de Plato:". There is an "Insertar" button below the form. Below the form, it says "Platos Disponibles" and "Tenemos gran cantidad de Platos". Below this is a table with two columns: "ID" and "NOMBRE".

ID	NOMBRE
1	Pollo con Salsa de Queso
2	Cangrejo con Arroz
4	Tallarines con Salsa y Carne
5	Arroz Chaufa y Pollo
6	Bistec a lo Pobre
7	Papas Salteadas con Escalopa

Figura 68: Ejemplo de mejora continua contemplada 1

- ✓ Agregar nueva funcionalidad al sistema, como, por ejemplo, Registrar datos del cliente.

The screenshot shows the same web interface as Figure 68, but now the form is filled with data. The "Nombre:" field contains "Roxana", the "Apellido:" field contains "Perez", the "telefono:" field contains "887766557", and the "Id de Plato:" field contains "5". The "Insertar" button is still present below the form.

Figura 69: Ejemplo de mejora continua contemplada 2

#### 4.14.2. Impacto post implementación

En la fase de post implementación o mejora continua se deben contemplar

las auditorias de la inversión, mejora continua del negocio y las actualizaciones del sistema en caso se requiera.

Para el caso de una auditoria de la inversión, el restaurant deberá evaluar el beneficio económico recibido como consecuencia de la implementación de nuestro Proyecto SoftCook Management.

Si nos enfocamos en la mejora continua del negocio, podemos establecer la integración de clientes con una plataforma más tecnológica, la simplificación de los procesos de negocio que mantiene el restaurant y así optimizar el uso de recursos que estos poseen, esto nos podría generar ciertos cambios o ajustes de la aplicación SoftCook Management vendida al restaurant.

Otra de las actividades que se contempla, es la gestión del conocimiento, capacitación y entrenamiento a usuarios finales, tanto desde el punto de vista transaccional como desde el punto de vista de generar algún tipo de reporte que dejemos configurado en la aplicación, estas actividades tienen como objetivo principal, lograr una apropiación y adopción de la herramienta otorgada.

## **5. Logros Obtenidos**

**Objetivo Específico.**

Desarrollar un Software mediante el cual se pueda integrar la cocina y la bodega de un restaurant.

### **Resultado Esperado.**

Dentro de las especificaciones funcionales del software se determinó que se necesitaba sincronizar los ingredientes de la preparación de platos de la cocina con el stock de estos ingredientes con la bodega. Así esta sincronización podría ayudar a que el encargado de la bodega revisara los ingredientes para determinar compras de éstos y también la ayuda a la cocina de no vender estos platos por falta de ingredientes.

Es por esto que la creación de SoftCook Management, viene a solucionar un inconveniente la que llevan muchos restaurantes pequeños y medianos los que no pueden pagar una licencia de un software que le ayude a la gestión de los platos, ingredientes y la bodega. Teniendo en cuenta que la administración de un software también conlleva capacitación a los integrantes del restaurant, por eso SoftCook Management posee una interfaz de usuario basada en la facilidad de uso para que el usuario o cliente que esté manejando desde el lado APP\_Cocina o App\_Stock no tenga problemas de aprendizaje nativo.

Siendo este proyecto una gran herramienta para la gestión de un restaurant de forma básica, se ha estudiado algunos elementos para complementar el desarrollo de esta herramienta para que sea integrada con otros participes dentro el mismo restaurant como el garzón (administración de mesas), facturación para la administración del restaurant y un cambio de tecnología a PWA<sup>12</sup> siendo este una actualización a los sistemas de información presentes actualmente.

## **6. Referencias**

---

<sup>12</sup> PWA: Progressive Web Apps. “Una aplicación web progresiva es un tipo de software de aplicación que se entrega a través de la web, creado utilizando tecnologías web comunes como HTML, CSS y JavaScript. Está destinado a funcionar en cualquier plataforma que use un navegador compatible con los estándares.” Información extraída de Wikipedia.

- PMOinformatica (2016). Modelo de estudio de Factibilidad de un proyecto consultado el 21 de marzo de 2020 en <http://www.pmoinformatica.com/2016/04/modelo-estudio-de-factibilidad.html>
- Mejía, M. (2009). Motor de Bases de Datos consultado el 21 de marzo de 2020 en <https://es.slideshare.net/triana25/motor-de-bases-de-datos-oracle>
- Validación de Datos: Definición, elementos y métodos. (2020). Retrieved 23 March 2020, from <https://www.tecnologias-informacion.com/validacion.html>
- <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>
- <https://es.slideshare.net/mrocio1/diseo-caso-de-pruebas-48099189>
- (2020). Retrieved 23 May 2020, from <http://winper.cl/Folletos/PresentacionInnovasoft%20v2.pdf>
- (2020). Retrieved 23 May 2020, from [http://opac.pucv.cl/pucv\\_txt/txt-4500/UCD4960\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-4500/UCD4960_01.pdf)
- <http://resultados.censo2017.cl/Region?R=R09>
- [https://www.resto.cl/?gclid=CjwKCAjw2uf2BRBpEiwA31VZj1oRYGkN3qHvT\\_ryK8WbxT4P20kpidqchXaDzjhIHXiHME89VJbcT2BoC9QIQAvD\\_BwE](https://www.resto.cl/?gclid=CjwKCAjw2uf2BRBpEiwA31VZj1oRYGkN3qHvT_ryK8WbxT4P20kpidqchXaDzjhIHXiHME89VJbcT2BoC9QIQAvD_BwE)
- <http://smartrestobar.cl/>

## **7. Conclusiones y Reflexiones**

Las conclusiones se encuentran divididas en dos secciones con la finalidad de presentar los resultados generales de este proyecto, estas son:

- Objetivo
- Objetivo Especifico

**Objetivo:**

Para finalizar la técnica de planificación es fundamental para el desarrollo del software ya que estas cobran mayor importancia en la medida que los proyectos son más complejos y requieren de mayores recursos. Si bien es cierto las creaciones de proyecto suelen ser optimizaciones de procesos, gracias a esto se aumenta la fluidez y se reduce el tiempo de uso, más aún cuando todo se basa en las nuevas tecnologías, por lo que eso le da un aporte positivo a la hora de crear un nuevo proyecto.

**Objetivo Específico:**

Complementando el objetivo se puede concluir que:

- La creación de los proyectos informáticos es mucho más efectiva.
- Para mejorar la gestión de cualquier negocio es esencial invertir en la adquisición de tecnologías que le permitan cumplir el objetivo principal.
- El uso de tecnologías Open Source incrementa la posibilidad de que empresas pequeñas y medianas puedan alcanzar la adquisición de nuestro producto.
- Sin el análisis de requerimientos no se puede crear un proyecto en base a las necesidades del restaurant.
- La participación del cliente en las etapas del proyecto fue fundamental para el éxito del proyecto.
- El mal uso del sistema indica una falta de conocimiento de los usuarios del sistema.
- Los fracasos de los proyectos son por una gestión mal realizada.