

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA
GESTIÓN DE PRODUCCIÓN EN EMPRESA DE CERVECERÍA
ARTESANAL APLICANDO KANBAN**

MORÁN JORQUERA WILSON MARTIN

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

DIRECTOR: ING. JAVIER CÓNDOR

QUITO, 2018

DEDICATORIA

A mi padre Wilson.

El ingeniero, el máster, mi modelo a seguir sin duda, quién siempre con el vivo ejemplo me demostró que con esfuerzo, perseverancia y muchas ganas, se puede salir adelante y lograr cumplir así las metas y sueños que uno se ha planteado. Sin él, no sólo mi vida universitaria y profesional serían diferentes, sino toda mi vida en general no sería tal y como lo es ahora. Por todo el valor mostrado para salir siempre adelante, por el esfuerzo que hace por su familia día a día, y por todo su amor.

A mi madre Verónica.

Mi querida madre, el auténtico y preciso ejemplo de que el amor verdadero existe; quién con mucha paciencia supo guiarme en este largo proceso, siendo aquella fuente de ánimo casi inagotable cada vez que me sentí cansado o perdido en algún punto. Por haberme apoyado en todo momento, por sus consejos, y su motivación constante. Por preocuparse en convertirme en una persona de bien; pero más que nada, por todo su amor.

AGRADECIMIENTO

En primer lugar quiero expresar mi más grande y sincero agradecimiento a mi tutor, el ingeniero Javier Córdor, quién con su conocimiento, capacidad de dirección y colaboración permitió el desarrollo de esta tesis. Por todo el soporte, el ánimo y la motivación brindada, y por estar siempre dispuesto a ayudar.

A los ingenieros Fabián de la Cruz y Damián Nicolalde, por involucrarse en el desarrollo de esta tesis y brindar siempre su colaboración para cada revisión. Por las precisas correcciones que realizaron; pero más que nada por los brillantes consejos que me compartieron siempre que fueron mis profesores. Me llevo un grato recuerdo de ellos.

A mis padres, quienes con esfuerzo y sacrificio me permitieron poder estudiar la carrera que quería en una prestigiosa universidad. Por todas las facilidades que me brindaron a lo largo de mi carrera. Por todo el apoyo incondicional, en verdad gracias, con su cariño y su paciencia, todo este proceso ha sido más sencillo.

Finalmente quiero expresar mi más sincero agradecimiento a todos esos amigos que siempre me apoyaron a lo largo de este duro proceso. Por todas sus palabras de aliento, sus ánimos, su apoyo. Significa mucho para mí contar en mi vida con gente que en verdad se preocupa por mi futuro y me acompaña en cada paso importante como este.

Tabla de contenido

Dedicatoria.....	
Agradecimiento.....	
Índice de Figuras.....	
Índice de Tablas.....	
Resumen.....	
Capítulo 1 Marco Teórico.....	
1.1. Cerveza Artesanal.....	
1.2. Metodología de Desarrollo KanBan.....	
1.3. Herramientas de Desarrollo.....	
1.3.1. Lenguaje de Programación.....	
1.3.2. Sistema Gestor de Base de Datos.....	
1.4. Arquitectura.....	
Capítulo 2 Situación Actual de la Empresa.....	
2.1. Historia.....	
2.2. Organigrama.....	
2.3. Proyección a Futuro.....	
2.4. Procesos de la Empresa.....	
2.4.1. Inventario.....	
2.4.2. Producción.....	

2.4.3. Distribución.....

2.4.4. Reporte Mensual.....

Capítulo 3 Análisis y Diseño.....

3.1. Análisis de Requerimientos.....

3.1.1. Gestión de Inventario.....

3.1.2. Producción de Cerveza.....

3.1.3. Distribución del producto.....

3.1.4. Gestión de Reporte Mensual.....

3.2. Definición de Requerimientos.....

3.2.1. Requerimientos Funcionales.....

3.2.2. Requerimientos No Funcionales.....

3.3. Modelo de Datos Conceptual.....

3.3.1. Módulo de Compras.....

3.3.2. Módulo de Ventas.....

3.3.3. Módulo de Producción.....

3.4. Diseño de Interfaces.....

3.4.1. Interfaz para CRUD básico.....

3.4.2. Interfaz para Consulta.....

3.4.3. Interfaz para Manejo de Orden.....

3.4.4. Interfaz para Manejo de Producción.....

3.4.5. Interfaz para Reportes.....

3.5. Diagrama de Casos de Uso.....

3.5.1. Siguiete Nivel: Administración de Productos.....

3.5.2. Siguiete Nivel: Gestión Órdenes de Producción.....

3.6. Diagrama de Paquetes.....

Capítulo 4 Desarrollo.....

4.1. Iteración 01.....

4.1.1. Puntos Desarrollados.....

4.1.2. Tablero KanBan.....

- 4.2. Iteración 02.....
 - 4.2.1. Puntos Desarrollados.....
 - 4.2.2. Tablero KanBan.....
- 4.3. Iteración 03.....
 - 4.3.1. Puntos Desarrollados.....
 - 4.3.2. Métricas KanBan.....
- 4.4. Iteración 04.....
 - 4.4.1. Puntos Desarrollados.....
 - 4.4.2. Tablero KanBan.....
- 4.5. Iteración 05.....
 - 4.5.1. Puntos Desarrollados.....
 - 4.5.2. Tablero KanBan.....

- Capítulo 5 Pruebas.....**
 - 5.1. Pruebas de Integración.....
 - 5.1.1. Prueba 01.....
 - 5.1.2. Prueba 02.....
 - 5.1.3. Prueba 03.....
 - 5.1.4. Prueba 04.....
 - 5.1.5. Prueba 05.....
 - 5.2. Pruebas de Aceptación.....
 - 5.2.1. Resultados.....
 - 5.2.2. Métricas KanBan.....
 - 5.3. Revisión KanBan Board.....

- Capítulo 6 Conclusiones y Recomendaciones.....**
 - 6.1. Conclusiones.....
 - 6.2. Recomendaciones.....

- Referencias.....

Índice de Figuras

Figura 1.1 Ejemplo de Tablero KanBan.....	
Figura 1.2 Java Virtual Machine.....	
Figura 1.3 Arquitectura de 3 Capas.....	
Figura 2.1 Organigrama de la Empresa.....	
Figura 2.2 Manejo de Inventario: Ingreso de Productos.....	
Figura 2.3 Manejo de Inventario: Salida de Productos.....	
Figura 2.4. Proceso de Producción.....	
Figura 2.5. Proceso de Contabilidad.....	
Figura 3.1 Datos Ejemplo de Orden de Compra.....	
Figura 3.2 Datos Ejemplo de Orden de Producción.....	
Figura 3.3 Datos Ejemplo de una Orden de Pedido.....	
Figura 3.4 Módulo de Compras.....	
Figura 3.5 Módulo de Ventas.....	
Figura 3.6 Módulo de Producción.....	
Figura 3.7 Modelo Conceptual de Datos.....	
Figura 3.8 Interfaz CRUD.....	

Figura 3.9 Interfaz Consulta
Figura 3.10 Interfaz Orden
Figura 3.11 Interfaz Producción
Figura 3.12 Interfaz Reportes
Figura 3.13 Diagrama General Casos de Uso
Figura 3.14 Caso de Uso Administrar Productos
Figura 3.15 Caso de Uso Gestionar Orden de Producción
Figura 3.16 Diagrama de Paquetes
Figura 4.1 Tablero KanBan 01
Figura 4.2 Tablero KanBan 02
Figura 4.3 Tablero KanBan 03
Figura 4.4 Tablero KanBan 04
Figura 4.5 Tablero KanBan 05

Índice de Tablas

Tabla 1.1 Características de KanBan
Tabla 3.1 Datos ejemplo de Materias Primas
Tabla 3.2 Datos ejemplo de Productos
Tabla 3.3 Datos ejemplo de Proveedores
Tabla 3.4 Datos ejemplo de Clientes
Tabla 5.1 Pruebas de Integración 01
Tabla 5.2 Pruebas de Integración 02
Tabla 5.3 Pruebas de Integración 03
Tabla 5.4 Pruebas de Integración 04
Tabla 5.5 Pruebas de Integración 05
Tabla 5.6 Pruebas de Aceptación

Resumen

La cerveza es una de las bebidas más antiguas y apreciadas en todo el mundo. Se dice que, por su exquisita espuma o color, puede ser al mismo tiempo causa y solución de muchos problemas. De hecho, su ingesta moderada puede traer múltiples beneficios al organismo por sus componentes, los cuales no son difíciles de adquirir. Además, el proceso a seguir para su elaboración está hoy en día muy bien detallado en diversos textos, sitios web, etc. Por esto y más, muchas personas en la actualidad se dedican a fabricar esta mítica bebida, nada más y nada menos que desde la comodidad de su casa.

Las personas que elaboran cerveza artesanal, por lo general, empiezan desde lo más básico. Un lugar pequeño, poco equipo, insumos básicos. El manejo de sus procesos se hace de una forma bastante manual; es decir se apoyan por lo general en el uso de pizarras, notas, post-it, y demás herramientas para manejar procesos como llevar control de su contabilidad, de su inventario, de su producción, o incluso su distribución.

El desarrollar un sistema de información que permita la gestión de producción en empresas de cervecería artesanal hace que este trabajo de disertación sea importante, ya que la mayoría de cervecerías artesanales no cuentan con uno que le permita gestionar sus procesos más básicos. A pesar de que ya existen algunos sistemas que podemos encontrar en la web, y que están destinados a cervecerías artesanales, pocos cuentan con licencias gratuitas, que es una de las preferencias de cualquier empresa que está emprendiendo. Por esto motivo se desarrolló el sistema para la cervecería de Caso de Estudio: Saint Roots Brewing Co.

De esta manera, se pretende facilitar la vida a los productores de cerveza artesanal; brindándoles un gestor de sus principales procesos como son el manejo de inventario y de la distribución de su producto, y principalmente el proceso de producción. Se pretende perfeccionar la administración de la cervecería, para mejorar así su eficacia y su eficiencia, permitiendo automatizar el manejo de alguno de sus procesos; de forma que su gestión se haga de manera más rápida, y que el acceso a la información de la empresa sea mucho más sencillo, controlado y veloz.

En primer lugar se hizo una investigación para adentrarnos un poco en el campo de la producción de cerveza artesanal. Fue necesaria la identificación de procesos que se

realizan en la cervecería, para recolectar toda la información referente a las necesidades del usuario. Con esto se pudo hacer un levantamiento de requerimientos, identificando todo aquello que se requería desarrollar. Se procedió luego a realizar un diseño del sistema, para así desarrollarlo de forma que cumpla con todo lo analizado. Finalmente, después de cada iteración, se realizaron las correspondientes pruebas de aceptación con el usuario para verificar su funcionamiento y obtener así el sistema deseado.

Capítulo 1 Marco Teórico

El marco teórico consiste en la recopilación de antecedentes y de las consideraciones teóricas necesarias para sustentar un proyecto. Es el soporte conceptual de todos aquellos conceptos teóricos que se utilizaron para el planteamiento y resolución del problema de un proyecto o de una tesis de investigación o desarrollo. De esta manera, es posible formular de una forma confiable, ordenada y coherente las conclusiones del proyecto.

En el presente marco teórico, se definen todos los conceptos relevantes que se profundizarán en el desarrollo del sistema de información. Se tratan temas importantes relacionados con la elaboración de cerveza artesanal y sus insumos, y además se define la metodología de desarrollo, la arquitectura, y las herramientas con las que el sistema de información será desarrollado.

1.1. Cerveza Artesanal

La cerveza es una de las bebidas más antiguas del mundo, junto con el vino. Desde hace miles de años, el ser humano ha venido disfrutando de todo tipo de cervezas. Es un producto muy apreciado por montones de personas. Sin embargo, la cerveza que más se produce, y se consume, es aquella conocida como cerveza industrial. Esta se produce con insumos más económicos y rendidores, por lo que les da a las cervecerías industriales un costo de producción bajo para producir en masa. De esta manera, son pocas las empresas que producen cerveza artesanal.

¿Pero qué significa entonces elaborar cerveza artesanal? La Sociedad Ecuatoriana de Cerveceros Artesanales afirma que: “Esto supone fabricar un producto bajo un proceso

totalmente controlado por el hombre, sin la utilización de químicos y respetando la pureza de dicho producto” (SECA, p.6).

El esfuerzo y preocupación que un cervecero artesanal pone en la elaboración de su producto no es para nada comparable con el de las cervecerías industriales, ya que no cuenta con toda la tecnología de las grandes empresas; es por esto que el producto final del cervecero artesanal es un producto mejor cuidado, y con un nivel de calidad superior.

A continuación, se muestran los principales insumos que se requiere para la elaboración de la cerveza artesanal:

- **Malta:** “La malta se obtiene de la cebada, la cual es una planta gramínea y está en la categoría de cereal, como el trigo o el maíz” (SECA, p.9). Es la que permite obtener diferentes colores y tipos de cerveza.
- **Lúpulo:** El lúpulo es una planta trepadora, que cumple funciones importantes como dar amargor, sabor y aroma a la cerveza; además de contribuir a su conservación.
- **Levadura:** La levadura es la responsable de generar el alcohol en la cerveza, además del gas carbónico. “La levadura es un micro-organismo, que se encuentra en estado de suspensión, es decir “dormido”” (SECA, p.13). Estos se alimentan de azúcar, y lo transforman en alcohol, generando al mismo tiempo gas carbónico.

Hay tantos estilos de cerveza, cada una con su sabor característico impuesto por la levadura utilizada, la malta, los lúpulos, o inclusive el tiempo de fermentación. Cambiar cualquiera de estos ítems puede resultar en un nuevo estilo de cerveza obtenido (Palmer, 2006). Es por esto que un sistema que permita gestionar la producción de cerveza

artesanal se vuelve importante, de forma que el cervecero pueda tener control sobre toda su producción.

1.2. Metodología de Desarrollo KanBan

En los últimos años, es notable la adopción de metodologías ágiles para el desarrollo de cualquier tipo de proyecto. Las metodologías ágiles nos permiten tener un producto final capaz de adaptarse de mejor manera a las necesidades del usuario, y haciendo uso de ellas, tenemos además la capacidad de adaptarnos al cambio.

Estas se basan en la entrega de pequeños prototipos evolutivos, añadiendo funcionalidades en cada una de las iteraciones, en las que es importante considerar la retroalimentación del usuario, ya que nos permite desarrollar el producto con un ciclo de vida iterativo e incremental según las necesidades del cliente.

KanBan se basa en un sistema de producción que dispara trabajo solo cuando existe la capacidad para procesarlo. Tiene sus orígenes en la empresa Toyota, donde se utilizaban tarjetas para identificar necesidades de material en la cadena de producción. KanBan es una palabra japonesa que puede traducirse como “tarjetas visuales”, donde Kan corresponde a *visual*, y Ban corresponde a *tarjeta*.

En la actualidad, KanBan ha pasado a formar parte de las llamadas metodologías ágiles de desarrollo. Ha sido usado en procesos de producción desde hace mucho tiempo, siendo considerado como un sistema de producción altamente efectivo y eficiente, contribuyendo a muchas empresas a generar un panorama de producción óptimo y competitivo.

Aunque KanBan ha sido aplicado en sistemas de producción de empresas y organizaciones desde hace mucho tiempo, no fue hasta 2004 que la metodología fue

introducida en el desarrollo de software por David Anderson de Microsoft, reemplazando las tradicionales tarjetas por un tablero visual muy similar a la metodología Scrum pero con características adicionales.

KanBan entonces, puede ser definido como un enfoque para el desarrollo ágil de software. “KanBan se basa en una idea muy simple: el trabajo en curso (Work In Progress, WIP) debería limitarse, y sólo deberíamos empezar con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena” (Kniberg & Skarin, 2014, p.7).

El principio de KanBan es que comiences con lo que sea que estés haciendo ahora mismo. El método KanBan, en el desarrollo de software, permite manejar equipos de desarrollo para visualizar el flujo de trabajo, limitar el trabajo en progreso (WIP) en cada etapa del flujo, y permite medir el tiempo de ciclo (lead time).

Con la aplicación de tres reglas fundamentales, KanBan demuestra ser una de las metodologías ágiles que menos resistencia al cambio presenta. Estas reglas son las siguientes:

1. Visualizar el flujo de trabajo

Esta regla consiste en la visualización de todo el proceso de desarrollo del proyecto, generalmente mediante el uso de un tablero físico. Este tablero se divide en columnas, las cuales representan un proceso de trabajo. Un ejemplo clásico en desarrollo de software sería dividirlo en columnas como: Análisis, Desarrollo, Pruebas.

La visualización del flujo de trabajo que KanBan nos ofrece por medio del tablero es muy útil. Este proporciona visibilidad del proceso de desarrollo de software, mostrando claramente el trabajo asignado para cada miembro del equipo, las fases de desarrollo del producto, las prioridades del trabajo, y muestra además donde se pueden producir

cuellos de botella, de manera que el equipo sepa donde concentrarse para resolver los problemas que bloquean el proceso y así restaurar el flujo productivo.

En la Figura 1.1, se presenta un clásico ejemplo de cómo llevar el control de tareas pendientes y terminadas en un tablero KanBan, dividido en fases de Análisis, Desarrollo y Pruebas respectivamente:

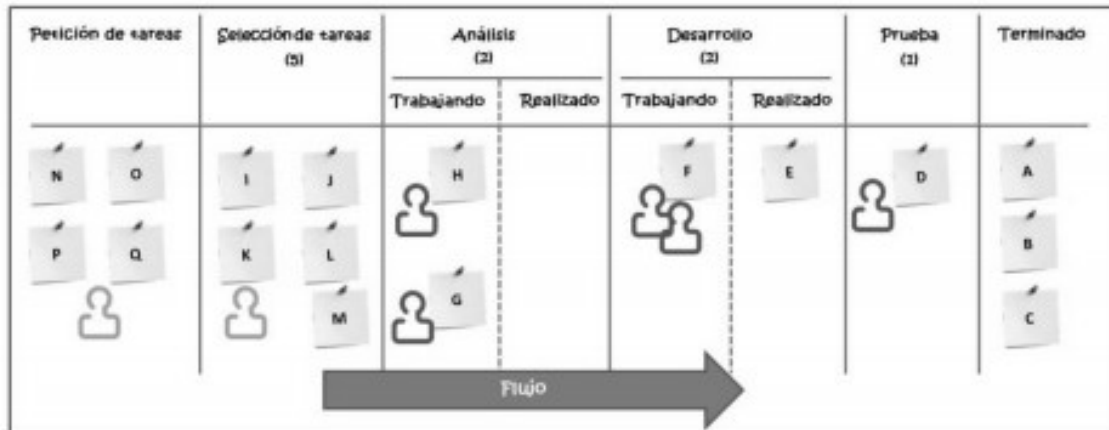


FIGURA 1.1 EJEMPLO DE TABLERO KANBAN.
(BONILLA HUERTA & RAMÍREZ CRUZ, 2014).

2. Limitar el Trabajo en curso

Los límites del *WIP* consisten en acordar con todo el equipo de manera anticipada, el total de ítems que pueden abordarse por cada proceso. Es decir, la cantidad de tareas que pueden estar en progreso durante cada fase de desarrollo (representado por cada una de las columnas del tablero). Uno de los principales objetivos de establecer estos límites es el de poder detectar cuellos de botellas.

KanBan de esta manera, limita el trabajo en curso (WIP) de acuerdo a la capacidad del equipo y de cada uno de sus miembros. Esto ayuda a visualizar los problemas del proceso, minimiza los defectos y mantiene un flujo estable. “Al limitar el trabajo en curso se consigue un ritmo de desarrollo sostenible, elevando la calidad de los productos

y un mayor rendimiento de los integrantes del equipo” (Bonilla Huerta & Ramírez Cruz, 2014, p.102).

3. Optimizar el flujo de trabajo

Uno de los objetivos de KanBan es hacer que la producción sea estable, previsible y continua. Esto se logra midiendo el tiempo que demanda un ciclo completo de ejecución, denominado *Cycle Time*.

El valor de nuestro rendimiento de trabajo, conocido también como *Throughput*, representa la cantidad de ítems que el equipo puede completar en un determinado período de tiempo. La optimización definitiva de nuestro flujo de trabajo consistirá fundamentalmente en la búsqueda de:

- Minimizar el Cycle time
- Maximizar el rendimiento de trabajo o Throughput
- Lograr que exista una variabilidad mínima entre Lead time y Throughput

El hecho de que se cuente con un flujo estable y calidad estable en el proceso de desarrollo, ayuda a reducir el tiempo de ciclo, logrando producir entregables de forma regular. Esto permite obtener retroalimentación del cliente o usuario con cada uno de estos entregables, de forma que se incrementa la confianza del mismo hacia la empresa que desarrolla el producto.

En resumidas cuentas, las principales reglas y características de KanBan se pueden visualizar de manera general en la Tabla 1.1:

Característica	Descripción
Visualizar el flujo de trabajo	Divide el trabajo en bloques, se escribe cada ítem en una tarjeta y se coloca en el tablero.
Limitar el trabajo en curso (WIP)	Asigna límites explícitos de cuantos ítems pueden estar en progreso en cada estado del flujo de trabajo.
Mide el tiempo de ciclo medio de un elemento (lead time)	Optimiza el proceso para hacer que el lead time sea tan pequeño y predecible como sea posible.

TABLA 1.1 **CARACTERÍSTICAS DE KANBAN.**
(MORÁN JORQUERA, 2018).

Sin lugar a dudas, KanBan es una herramienta sumamente útil para cualquier tipo de proyecto, incluso en los de desarrollo de software, gracias a todas las características, beneficios y ventajas que ofrece sobre otras metodologías. “KanBan expone los cuellos de botella, colas, variabilidad y desperdicios. Todas las cosas que impactan al rendimiento de la organización en términos de la cantidad de trabajo entregado y el ciclo de tiempo requerido para entregarlo” (Kniberg & Skarin, 2014, p.8).

1.3. Herramientas de Desarrollo

1.3.1. Lenguaje de Programación

El desarrollo del sistema de información se lo realizará con el lenguaje de programación Java. El lenguaje de programación orientada a objetos Java, es creado por Mike Sheridan, Patrick Naughton y James Gosling, de la empresa Sun Microsystems a inicios de la década de 1990. Java incorpora diversos aspectos dentro del propio lenguaje, y es por esto que es considerado el lenguaje ideal para aprender (Flórez Fernández, 2012).

Su objetivo es convertirse en el lenguaje universal de desarrollo, haciendo uso de su JVM. “Java Virtual Machine, JVM o máquina virtual de Java, es un proyecto que se puede instalar en cualquier sistema operativo y permite que un archivo compilado se ejecute sin requerir cambios, independientemente del procesador que posea el computador” (Flórez Fernández, 2012, p.4).

Esta máquina viene a ser nada menos que el intérprete de Java. De esta manera, nos evitamos tener que desarrollar un programa diferente para cada CPU o para cada plataforma.

El compilador de Java realiza un análisis del código escrito en archivos fuente de Java con extensión “.java”, y si no encuentra errores, genera un archivo compilado con extensión “.class”. Luego, lo que hace la JVM es ejecutar estos archivos compilados; donde cada sistema operativo cuenta con su propio JVM para hacerlo.

El funcionamiento de la JVM puede ser visualizado de mejor manera con la Figura 1.2:

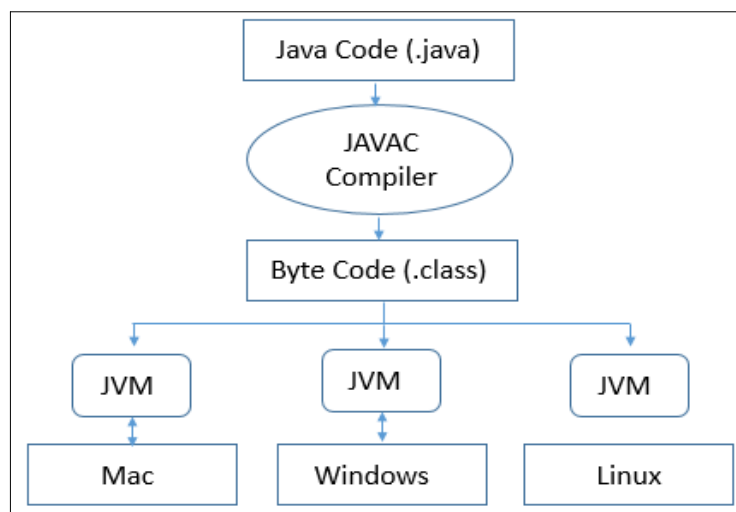


FIGURA 1.2 JAVA VIRTUAL MACHINE.
(NET-INFORMATIONS.COM, 2017).

Java es un lenguaje de objetos que incorpora el uso de la orientación a objetos como base fundamental del lenguaje, y es por esto que la programación orientada a objetos es el paradigma a utilizar en el desarrollo de este trabajo. La programación orientada a

objetos es definida como un paradigma que permite realizar una abstracción de la realidad, siendo capaz de ser implementado en una aplicación software con la finalidad de resolver problemas tomando como base problemas y objetos de la vida cotidiana.

Un primer acercamiento intuitivo a la programación orientada a objetos pasa justamente por los objetos, llamados también instancias de clase. “Un objeto representa un concepto dentro de un programa y tiene la información necesaria para abstraerlo: los datos que describen su estado y las operaciones que pueden modificar este y determinan las capacidades del objeto” (López, 2014, p.105).

Sin embargo, para la creación de objetos es necesaria la creación de clases. Las clases son tipos de datos que definen los atributos y los métodos que serán comunes para todos los objetos de dicha clase (López, 2014). Es decir, las clases son abstracciones que representan objetos que comparten un comportamiento e interfaz común.

Para una buena práctica de la programación orientada a objetos, es necesario aplicar un enfoque modular, que busca aplicar principios de abstracción al dividir un problema complejo en distintos módulos o partes.

Es por esto que para desarrollar el programa propuesto, se pretende trabajar con el término de modularidad. “La modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí, y de las partes restantes” (Ruiz, 2011, p.10).

De esta manera, es fácil darse cuenta todas las ventajas con las que contamos al desarrollar el sistema usando un paradigma orientado a objetos. El lenguaje orientado a objetos fue diseñado bajo un esquema en el que un programa escrito en dicho lenguaje,

podiera ejecutarse en cualquier máquina, es decir, no solo en computadoras, sino en una diversa cantidad de dispositivos electrónicos.

1.3.2. Sistema Gestor de Base de Datos

Para el desarrollo del sistema de información, se hará uso de una base de datos relacional. La información que se almacena en estas bases es percibida por el usuario como una serie de tablas, y los datos alojados en estas tablas pueden ser además manipulados por el usuario. Las tablas son las estructuras más importantes dentro de una base de datos relacional.

Es fundamental que los datos se organicen de forma que su gestión se facilite, para que de esta manera el acceso a los mismos sea rápido y eficaz. Es gracias a los sistemas gestores de base de datos, que es posible realizar la gestión de tablas de manera sencilla, facilitando operaciones de creación, actualización, consulta, y eliminación de datos.

“Los sistemas de gestión de bases de datos o SGBD (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan” (Valderrey Sanz, 2014, p.37). También llamados motores de base de datos, son una colección estructurada de datos relacionados entre sí, con un conjunto de programas que permiten acceder y gestionar los mismos.

La transaccionalidad y la seguridad de datos son características importantes de un motor de base de datos. Una transacción es la unidad de ejecución de un programa que accede a los datos de una base y que, por lo general, los actualiza (Camuña, 2014). Cuando existe más de un usuario con acceso a la base de datos, es necesario que el SGBD

asegure que las acciones que realizan estos usuarios no interfieran entre sí; además de ser capaz de recuperarse de manera correcta de los fallos o errores que se presenten.

De esta manera, al usar un sistema gestor de base de datos, cuando exista la posibilidad de corrupción de la base de datos, se emplea un mecanismo de transacciones con el que se elimina toda fuente de dicha corrupción en la base. Así, la base de datos no quedará en un estado inconsistente, garantizando la seguridad e integridad de los mismos.

En la actualidad existen diversos gestores de bases de datos que permiten la creación y gestión de bases de datos, así como el manejo de las estructuras necesarias para el almacenamiento y búsqueda de la información que requerimos de la manera más eficiente posible. Entre ellos, vale la pena destacar algunos que son los más usados hoy en día, la mayoría relacionales:

- MySQL
- Microsoft SQL Server
- PostgreSQL
- Oracle

1.4. Arquitectura

Desarrollar una aplicación basada en una arquitectura, permitirá que la misma cuente con algunos aspectos fundamentales del paradigma de orientación a objetos, como son bajo acoplamiento, alta cohesión, e incluso escalabilidad. Una arquitectura básicamente define un conjunto de capas, en donde cada una se especializa en proveer distintos servicios para resolver diversos tipos de problemas o comportamientos de la aplicación.

Para el desarrollo del sistema de información deseado, se hará uso de un lenguaje de programación que permite el esquema de 3 capas: Modelo, Vista, Controlador.

La arquitectura de tres capas es una técnica en el desarrollo de aplicaciones de software que tiene como objetivo la separación de la lógica del negocio de la

presentación y de la persistencia. Una de las principales ventajas se obtiene con el bajo acoplamiento de las aplicaciones debido a que esta característica, permite fácilmente realizar cambios en los servicios sin tener que revisar todos los componentes de la aplicación (Flórez Fernández, 2012, p.151).

Esta división en componentes ofrece muchas ventajas; entre ellas está el reducir la complejidad del código, agilizando así notablemente el proceso de desarrollo de software. El bajo acoplamiento que se logra con esta arquitectura permite que se realicen fácilmente cambios en los servicios, sin necesidad de revisar todos los componentes de la aplicación. Permite además distribuir el trabajo del equipo, donde cada equipo puede hacer uso de los componentes desarrollados por otro sin necesidad de conocer el desarrollo de los mismos. Se basa en el modelo de la Figura 1.3:



FIGURA 1.3 **ARQUITECTURA DE 3 CAPAS.**
(FLÓREZ FERNÁNDEZ, 2012).

1. **Capa de Presentación:** Esta capa exhibe la aplicación al usuario y muestra la información; además captura la información del mismo para comunicarse con la capa de lógica de negocio.
2. **Capa de Lógica de Negocio:** En esta capa es donde se desarrollan los algoritmos propios de la aplicación. De esta manera, provee servicios a la capa

de presentación recibiendo la información del usuario como parámetros, para luego comunicarse con la capa de persistencia.

3. **Capa de Persistencia:** En esta capa es donde se almacenan los datos, además de manejar todas las operaciones para la manipulación de dicha información.

Uno de los beneficios clave de aplicar esta arquitectura en el proyecto, es que nos permite realizar grandes modificaciones en el mismo, sin necesidad de alterar los procesos de la aplicación. Por ejemplo, si desarrollamos una aplicación para escritorio y en un futuro se desea usar esta aplicación para web, bastaría con modificar la capa de presentación para un ambiente web dejando las otras capas de la aplicación intactas. De igual manera, si se desea usar otro sistema gestor de base de datos, bastaría con modificar únicamente la capa de persistencia.

Para finalizar, esta arquitectura genera además algunos beneficios clave para el proyecto a desarrollar, de las cuales se pueden destacar:

- ✓ Escalabilidad
- ✓ Portabilidad
- ✓ Usabilidad

Capítulo 2 Situación Actual de la Empresa

La situación actual de una empresa brinda un claro entendimiento del entorno en donde actúa la misma, además de mostrar sus principales fortalezas y debilidades. Así, es posible conocer posibles mejoras que se pueden adoptar para la mejora constante de la misma. Este proceso de conocer la situación actual de la empresa, permite tener claro los pasos a seguir para tener éxito en el desarrollo del sistema deseado.

El presente capítulo tiene como objetivo mostrar la situación actual de la cervecería Saint Roots, detallando de manera general aspectos como su historia, su visión, y sus principales procesos. De esta manera, se pretende que el sistema de información se adapte a esta situación y facilite la administración de la misma, con el objetivo de alcanzar el éxito empresarial.

2.1. Historia

La empresa Saint Roots Brewing es una cervecería artesanal fundada en mayo del 2016. Esta nace como una idea de dos jóvenes que decidieron hacer realidad uno de sus mayores sueños, que era el de poder fabricar su propia cerveza, y así poder disfrutarla y compartirla con sus amigos y familia. Es así, que deciden asociarse y unir sus habilidades y conocimientos para poder fabricar una cerveza de calidad, con el objetivo de posicionarse como una de las mejores marcas de cerveza artesanal dentro del país.

La idea nace años atrás, cuando uno de los dos socios cursaba sus estudios en la carrera de ingeniería en alimentos en la universidad. Esta le brindaba todos los instrumentos y conocimientos necesarios para poder fabricar una diversidad de alimentos y bebidas; y

es aquí donde se da cuenta que era posible poder fabricar su propia cerveza. De esta manera, comienza a producir pequeños lotes de cerveza para sus proyectos universitarios, compartiéndolas además con familia, amigos y conocidos para obtener la mayor retroalimentación posible sobre la calidad del producto.

Una vez graduado como ingeniero en alimentos, decide compartir esta idea con su gran amigo de toda la vida, para asociarse y empezar a fabricar cerveza artesanal como empresa independiente. Montan una pequeña planta en el norte de la ciudad de Quito, y adquieren todos los insumos y equipos necesarios para la producción. Finalmente, su nuevo socio diseña lo que sería la imagen de la empresa y de esta manera deciden lanzar su producto al mercado ecuatoriano.

Saint Roots, con más de dos años de trayectoria, se ha convertido en una microcervecería que se dedica a compartir su producto y su cultura cervecera con el Ecuador y con el mundo. Con una pasión por la cerveza que los distingue de otras cervecerías, fabrican un producto de calidad, digno de ser compartido en grandes momentos, ya que experimentan con ingredientes increíbles que a cualquiera le costaría imaginar que pueden mezclarse con la cerveza, incluyendo sabores típicos del Ecuador.

De esta manera, son capaces de crear cerveza no solo apta para cerveceros y expertos de la cultura cervecera, sino para cualquier persona que gusta de experimentar cosas nuevas y diferentes. La etiqueta e imagen es personalizada para cada una de las cervezas que producen, lo que hace fácil que uno se identifique con su favorita. La empresa no sólo busca brindar un producto, sino brindar una experiencia diferente a sus clientes.

Mezclando sabores tradicionales y modernos, y con diseños totalmente innovadores y diferentes, la empresa ha logrado posicionarse muy bien en el mercado ecuatoriano, logrando ganar distintos reconocimientos y galardones dentro del mundo cervecero. De

esta manera, el gran objetivo es ser reconocido como una de las mejores marcas de cerveza artesanal en el país.

2.2. Organigrama

La empresa en la actualidad cuenta con dos socios, quienes se encargan de la dirección y administración de la misma. La empresa se dividió en dos departamentos principales: el Departamento Comercial y el Departamento de Operaciones. Cada uno de los socios de la cervecería es, además de director, gerente de uno de estos departamentos y está encargado de todas las actividades involucradas.

Estas actividades incluyen los procesos fundamentales para la operación de la empresa en el mercado, como son:

- Marketing
- Contabilidad
- Distribución del producto
- Ventas / Manejo de Clientes
- Producción
- Bodegaje
- Calidad
- Compras / Manejo de Proveedores

Cada uno de los departamentos y sus procesos involucrados, son detallados en el organigrama de la empresa, mostrado en la Figura 2.1:

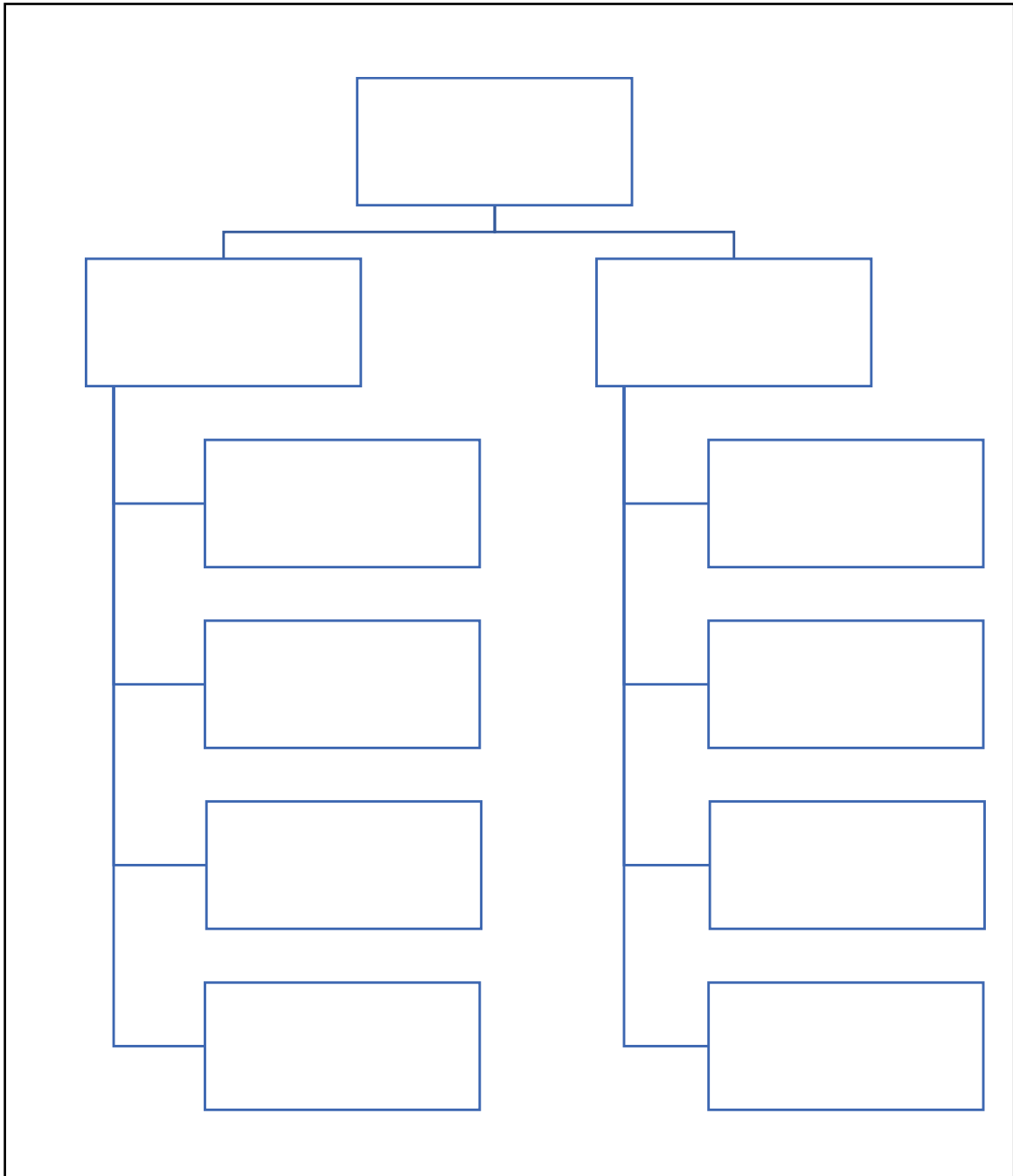


FIGURA 2.4 **ORGANIGRAMA DE LA EMPRESA.**
FORMADA POR DOS DEPARTAMENTOS PRINCIPALES EN DONDE CADA UNO DE LOS SOCIOS SE ENCARGA DE LA GERENCIA Y ACTIVIDADES INVOLUCRADAS EN LOS MISMOS.
(MORÁN JORQUERA, 2018).

2.3. Proyección a Futuro

Proyectar el futuro de una empresa no es una tarea sencilla. Manejarla y sostenerla en el mismo nivel requiere llevar una armonía y equilibrio en todas las áreas de la misma. Una buena manera de proyectarse a futuro es establecer claramente una visión de la empresa, de manera que todos los miembros trabajen orientados a la misma y sepan que es lo que la empresa quiere llegar a ser.

La visión de la empresa es una manifestación que indica de manera clara hacia dónde se dirige la misma, y en qué es en lo que pretende convertirse a largo plazo. La visión de la empresa permite enfocar los esfuerzos de todos los miembros hacia una misma dirección, de manera que permite establecer objetivos y así formular estrategias para ejecutarlas, logrando una total coherencia y organización dentro de la empresa.

De esta manera, la empresa Saint Roots ha establecido una proyección a futuro enfocada en diferentes ámbitos, describiendo lo que necesita lograr de modo que se consiga la máxima calidad, sostenibilidad y crecimiento como cervecería dentro del mercado ecuatoriano. La proyección a futuro viene dada por los siguientes ámbitos:

- ✓ Personas: “Ser la cerveza artesanal favorita en el Ecuador, de forma que esté presente en esos grandes momentos para compartir, momentos necesarios para que las personas puedan dar así lo mejor de sí mismas cada día”.
- ✓ Bebidas: “Ofrecer una diversidad de sabores y productos de calidad, que satisfagan los deseos y necesidades de los más exigentes y atrevidos consumidores que siempre buscan nuevas experiencias”.

- ✓ Cultura: “Desarrollar y alimentar una red de trabajo exitosa tanto de clientes como de proveedores, para crear un valor común y aportar para que la cultura cervecera y el gusto por la cerveza artesanal siga creciendo en el país”.

- ✓ Productividad: “Ser una organización dinámica y eficaz, que reaccione rápidamente al cambio, de forma que se logre maximizar el rendimiento para los socios de la empresa”.

La empresa Saint Roots lo que pretende con todos estos objetivos enfocados a diferentes ámbitos, es proyectar una imagen positiva ante sus clientes, proveedores y ante todo el público en general. De esta manera, los socios de la empresa quieren mantener la identidad y personalidad de la empresa intactas.

La mejor innovación es la renovación permanente de todos los productos fundamentales de la marca, adaptando la imagen y sabores de la cerveza a los gustos actuales de los consumidores. El objetivo es innovar con éxito, aportando valor tanto al proveedor como al consumidor, para acelerar el crecimiento y asegurar el futuro.

2.4. Procesos de la Empresa

Las organizaciones son tan eficientes como lo son sus procesos. “Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido” (Maldonado, 2011, p.1).

Es decir, un proceso es la manera en la que la empresa hace las cosas. El fin de estas actividades va desde la producción de un bien o servicio, hasta la realización de una actividad interna de la organización. Como ejemplos clásicos dentro de una empresa

tenemos el proceso de producción, facturación, etc. Los procesos tienen que estar gestionados correctamente para obtener una alta eficiencia de los mismos.

La empresa maneja cuatro procesos principales, los cuales involucran todas las actividades principales para el funcionamiento de la empresa. Estos procesos son básicamente los siguientes:

1. Manejo de inventario
2. Producción
3. Distribución del producto
4. Reporte mensual de flujo de caja

A continuación, se explicará en detalle estos procesos que maneja la empresa.

2.4.1. Inventario

El manejo de inventario dentro de una organización es una parte fundamental para el éxito en la administración de la misma. Se entiende por inventario todos los recursos materiales que la empresa tiene almacenados, tanto aquellos que se utilizan en el proceso de producción (también llamados insumos), como también el producto terminado. La gestión de este proceso de manejo de inventario es una de las actividades clave en la dirección de las empresas productoras.

El inventario dentro de la empresa Saint Roots es manejado por el departamento de operaciones. El encargado del departamento cumple dos funciones en el proceso:

1. Registrar en el inventario de la empresa la entrada de todos los insumos, materia prima y herramientas que ingresan, una vez que se hayan recibido todos aquellos que estaban detallados en la orden de compra al proveedor. En la Figura 2.2 se muestra gráficamente el proceso de ingreso de insumos.
2. Registrar en el inventario de la empresa la salida de los productos terminados, una vez se hayan despachado todos aquellos que estaban detallados en la orden

de pedido del cliente. En la Figura 2.3, por su parte, se muestra gráficamente el proceso de salida de productos.

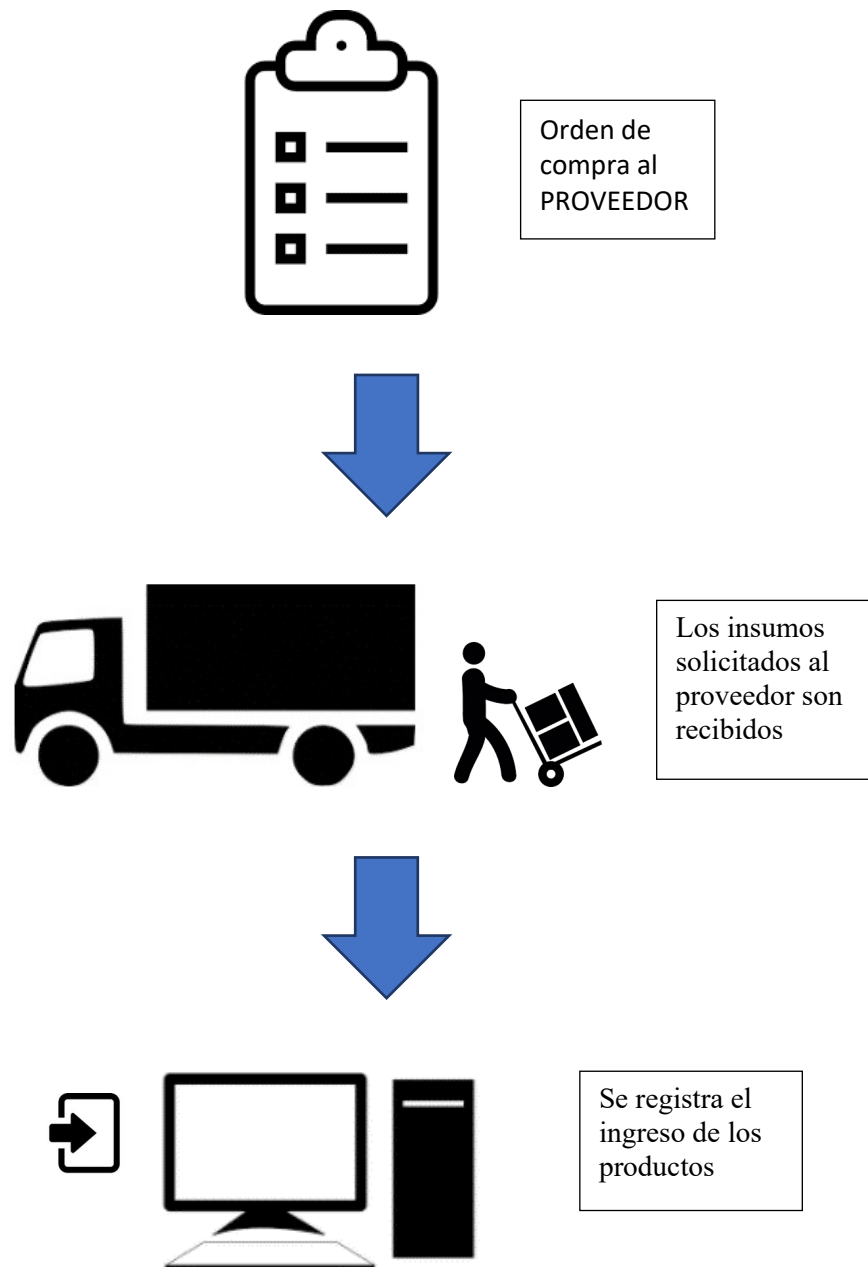


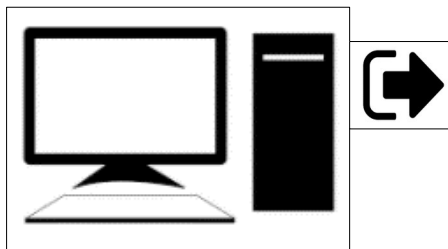
FIGURA 2.5 **MANEJO DE INVENTARIO: INGRESO DE PRODUCTOS.**
(MORÁN JORQUERA, 2018).



Orden de
pedido del
CLIENTE



Los productos
solicitados por
el cliente son
despachados



Se registra la
salida de los
productos

FIGURA 2.6 **MANEJO DE INVENTARIO: SALIDA DE PRODUCTOS.**
(MORÁN JORQUERA, 2018).

2.4.2. Producción

La producción de cerveza artesanal implica fabricar una cerveza bajo un proceso que es controlado en su totalidad por el hombre. Esto implica que no se utilizan conservantes ni ningún compuesto químico, ya que dañaría la naturalidad propia de la cerveza y su pureza. Este proceso de producción natural es el que marca la diferencia de una cervecería artesanal con el de una industrial.

Es por esto que los ingredientes que maneja la empresa Saint Roots en la producción de sus cervezas son solamente la malta de cebada y trigo, el lúpulo, la levadura y el agua. Es con estos 4 ingredientes que se realiza el proceso de elaboración de todas sus cervezas. En ciertas ocasiones, un ingrediente natural es añadido a la receta, lo que le otorga los diferentes sabores característicos a la cerveza, siendo este el valor añadido que la empresa ofrece al cliente.

La producción de la cerveza artesanal en la empresa Saint Roots es manejado por el departamento de operaciones. El proceso puede ser explicado de una manera muy general con los siguientes pasos:

1. **Maceración:** Según la RAE, la maceración es: “Mantener sumergida alguna sustancia sólida en un líquido a la temperatura ambiente, con el fin de ablandarla o de extraer de ella las partes solubles” (Real Academia Española, 2001). El objetivo es obtener de la malta la mayor cantidad de extracto. Para esto, se mezcla **la malta** molida con agua a altas temperaturas, dependiendo del tipo de cerveza a elaborar.

- 2. Lupulización:** El lupulado del mosto que se obtuvo en la fase anterior, consiste en añadir **el lúpulo** al mosto en ebullición. Dependiendo de la cantidad y de la variedad de lúpulo que se utilice, la cerveza tendrá un mayor o menor amargor, sabor y aroma. Se añade distintas variedades de lúpulo, en diferentes momentos de la ebullición según la cerveza deseada.
- 3. Fermentación:** El mosto enfriado que se obtuvo en la fase anterior, se envía a alguno de los tanques de fermentación, a la vez que se oxigena para permitir el crecimiento de la levadura. **La levadura** es añadida al fermentador para iniciar el proceso de fermentación, que consiste en la transformación de los azúcares del mosto en alcohol y anhídrido carbónico. Al finalizar, las levaduras se depositan en el fondo del fermentador y son retiradas para que su descomposición no altere el sabor de la cerveza. El proceso de fermentación dura entre 5 y 20 días.
- 4. Embotellado y Gasificación:** La cerveza obtenida, se embotella y se guarda en una sala con temperatura controlada para que se produzca una tercera fermentación que transformará los azúcares en gas carbónico, responsable de las burbujas y el gas característico de la cerveza.
- 5. Bodegaje:** Una vez ha transcurrido el tiempo de guarda de las cervezas, se procede a etiquetarlas según el tipo de cerveza que se produjo. Almacenadas en cajas, se colocan en la bodega fría para su posterior venta y distribución.

Los pasos generales a seguir para el proceso de producción de la cerveza artesanal pueden ser explicados de manera gráfica en la Figura 2.4:

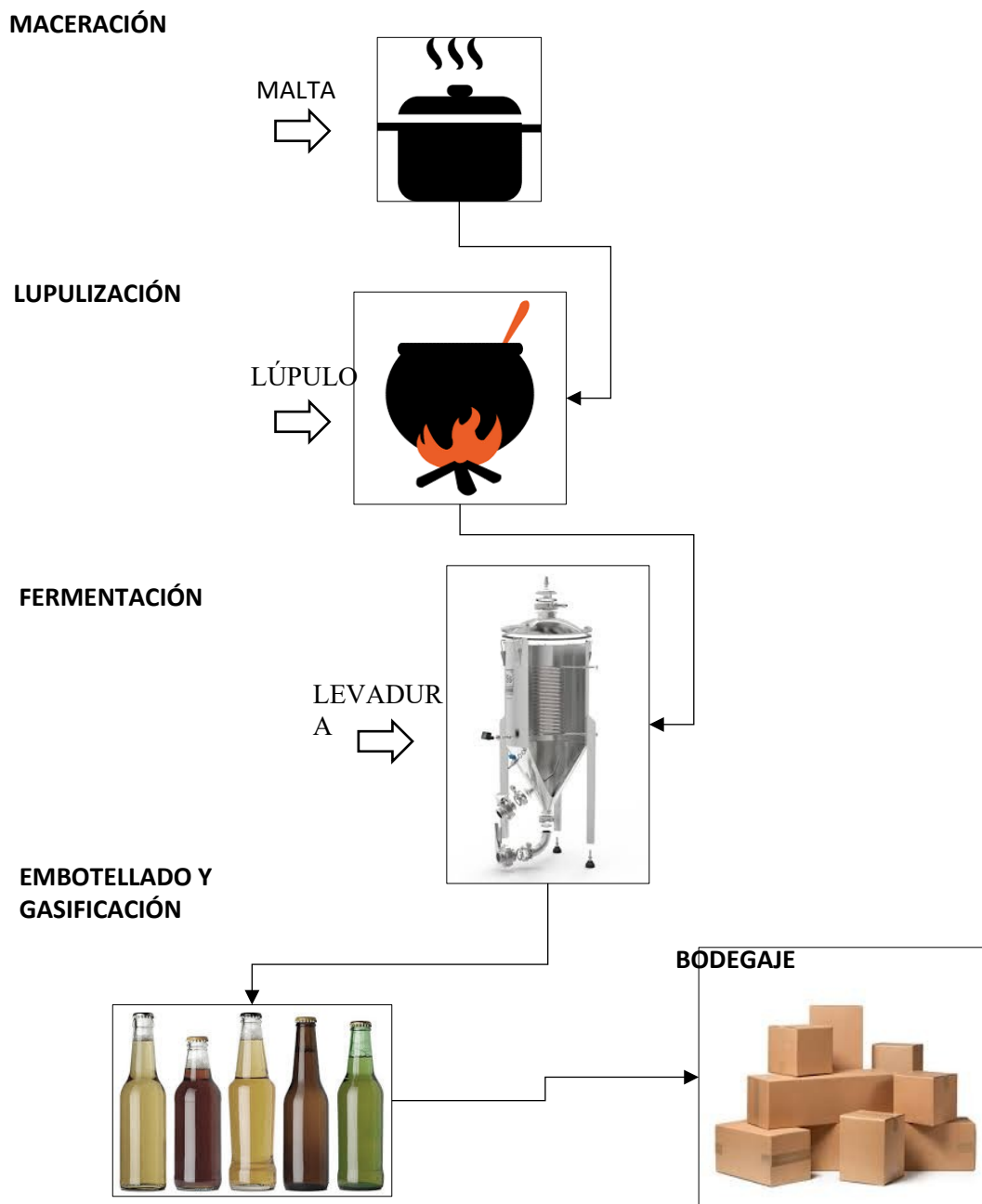


FIGURA 2.7. **PROCESO DE PRODUCCIÓN.**
(MORÁN JORQUERA, 2018).

2.4.3. Distribución

Antes de que un producto se encuentre listo para su introducción en el mercado, la empresa debe contar con un proceso de control de calidad, de manera que se asegure que el producto cumple con todas las normas necesarias. Una vez que se ha determinado la calidad, este se encuentra listo para su venta, por lo que es necesario que la empresa determine métodos para hacer llegar el producto a sus clientes.

El objetivo de Saint Roots es garantizar que el producto llegue al consumidor en el momento y lugar que lo precise, asegurando que sea distribuido en condiciones óptimas. Todo este proceso de distribución de cerveza artesanal incluye además elegir aquellos establecimientos que se acomoden a las necesidades del cliente, y que además cumplan con la política y cultura que la empresa tiene establecidos.

En esta empresa, una vez que el producto terminado almacenado en bodega ha pasado por los respectivos controles de calidad, se encuentra listo para su distribución. En este momento, el departamento comercial se encarga de la distribución del producto a aquellos clientes que han realizado pedidos a la empresa. Es fundamental asegurar la existencia del producto antes de empezar la distribución del pedido.

El proceso de distribución que se realiza dentro de la empresa puede ser explicado de manera muy general con los siguientes pasos:

1. El encargado de distribución recibe un pedido de un cliente. En el caso de ser un cliente nuevo, registra los datos del mismo, incluida la dirección de entrega preferida por el mismo.
2. Se crea una orden de pedido para el cliente y se registran todos los productos que el cliente solicita a la cervecería, además de la cantidad exacta de cada uno que el cliente requiere en su orden.

3. Se guarda la orden de pedido, y posteriormente se comprueba que haya suficiente stock de los productos que han sido solicitados en la orden. En base al stock actual del producto en inventario, se planifica y garantiza una fecha de entrega de la orden al cliente.
4. Cuando todos los productos en la orden de pedido se encuentren listos para ser entregados, estos se despachan en los vehículos repartidores de la empresa, siendo distribuidos a la dirección indicada por el cliente.
5. Finalmente, se controla con el cliente el estado de sus pedidos, para ver si estos han sido distribuidos con éxito o si presentan algún tipo de novedad, demora o inconveniente en el camino.

2.4.4. Reporte Mensual

La correcta gestión de la contabilidad dentro de una empresa es un proceso esencial para cualquier tipo de organización, ya que gracias a ella es posible llevar registro de todos los gastos, ingresos y transacciones que se realizan en general. Con esto, incluso se pueden realizar proyecciones a largo y corto plazo, siendo capaces de evitar muchas crisis e imprevistos que perjudican a la empresa y a sus socios.

“La contabilidad tiene por objetivo captar, procesar y transmitir la información adecuada al proceso de toma de decisiones de los múltiples usuarios de la misma. Para ello se lleva a cabo el proceso contable” (“El Proceso contable de una empresa”, 2016). Es por esto que el encargado de llevar el proceso de contabilidad, debe de introducir la totalidad de las operaciones económicas hechas por la organización. Llevando a cabo un proceso transparente, los miembros de la organización siempre tendrán la posibilidad de ver y entender la situación actual de la empresa, y de esta manera, el proceso de toma de decisiones será más acertado.

El objetivo del proceso de contabilidad dentro de esta empresa es poder reflejar en todo momento el flujo de caja de la misma. Este flujo de caja brinda toda la información necesaria con respecto a los movimientos de efectivo de la organización, indicando además los egresos, ingresos y los fondos disponibles hasta la fecha.

El proceso de contabilidad dentro de la empresa Saint Roots es manejado por el departamento comercial. El encargado en este caso, realiza dos tareas básicas:

1. Una vez que el cliente ha recibido su orden de pedido y la ha pagado, es necesario registrar este ingreso de dinero en la organización. El total del ingreso se calcula según los productos y la cantidad de los mismos que constaban en la orden de pedido del cliente y que fueron ya entregados. Para cada transacción que se realiza, se registra el número de factura o documento correspondiente que respalde dicho ingreso en la organización. Ver Figura 2.5.
2. Cuando se realiza una orden de compra de insumos y material al proveedor, o en manera general cada vez que se compra cualquier tipo de material para el uso de la empresa, es necesario desembolsar una cantidad de dinero. Es necesario registrar esta salida de dinero en la organización. Para cada transacción que se realiza en la compañía, se registra el número de factura o documento correspondiente que respalde dicho egreso en la organización. Ver Figura 2.5.

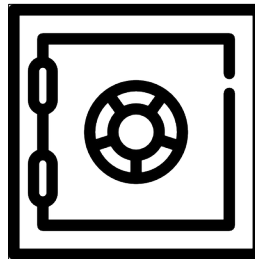
Todo ingreso y salida de dinero de la empresa es controlado, y respaldado además con un número de documento. De esta manera, es posible llevar un control más eficiente del flujo de caja de la empresa, brindando la posibilidad a los directivos de ver y entender la situación financiera actual de la empresa cada que vez que lo requieran. Basándose en este estado financiero, los directivos son capaces de tomar mejores decisiones a futuro y anticiparse a cualquier imprevisto.

1. Ingreso de Dinero



Se registra el ingreso, y se actualiza el estado financiero en la empresa

FLUJO DE CAJA



Se registra el egreso, y se actualiza el estado financiero en la empresa

2. Salida de Dinero

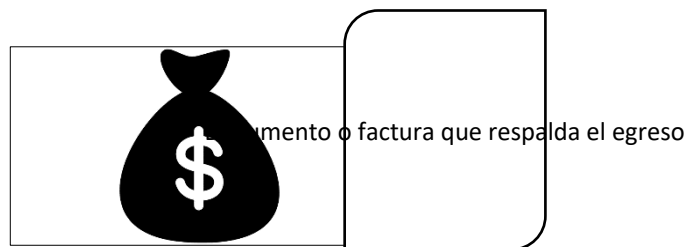


FIGURA 2.8. **PROCESO DE CONTABILIDAD.**
(MORÁN JORQUERA, 2018).

Capítulo 3 Análisis y Diseño

El desarrollo del sistema propuesto, al ser una herramienta que pretende ser utilizada en un futuro dentro del contexto de un problema real, tiene que seguir un disciplinado proceso de análisis y diseño que proporcione los respectivos cimientos bajo los cuales se va a desarrollar la aplicación. De esta manera, todas las etapas del desarrollo del sistema se vuelven más coherentes y formales.

Para que el desarrollo del sistema concluya de manera exitosa, es fundamental realizar el proceso de análisis en primer lugar para tener una completa y plena comprensión de los requerimientos. El proceso de diseño, por su lado, permite traducir estos requerimientos en una representación de software. En este capítulo, se detallan los procesos de ingeniería de software, análisis y diseño involucrados en el desarrollo del sistema de información.

3.1. Análisis de Requerimientos

El análisis de requerimientos del sistema de información que se desarrolló para la empresa, inicia con el respectivo levantamiento de información en las instalaciones de la cervecería. Esto incluye la identificación de procesos que se realizan en la misma, a través de la conversación con los administradores, de manera que se logre recolectar toda la información referente a las necesidades del usuario. El usuario del sistema, en este caso, son los directivos de la cervecería Saint Roots.

Finalmente, con toda esta información, se realizó un análisis para llevar a cabo el respectivo diseño del sistema de información. Este diseño incluye los diagramas de casos de usos para los procesos principales, y el diagrama de clases. De esta manera, se ha recolectado la información referente a las necesidades del usuario para cada uno de

los principales procesos que son: Gestión de Inventario, Gestión de Producción, Gestión de la Distribución y Gestión de Reportes de Ingresos vs Egresos.

3.1.1. Gestión de Inventario

La empresa maneja dos tipos de clasificaciones para manejar su inventario. En primer lugar está la materia prima que adquiere para la producción de su cerveza, donde podemos destacar productos como la malta, el lúpulo y la levadura. Por otro lado, una vez culminado el proceso de producción, la empresa maneja el producto terminado como tal que es la cantidad total de determinada cerveza que se obtuvo en el proceso.

Por lo anterior, la empresa requiere manejar la información de todos los **productos** que existen en su inventario. Esta información incluye:

- **Nombre:** un nombre que identifique a cada producto o materia prima.
- **Costo:** costo al que se adquirió la materia prima.
- **Precio:** precio al que se venderá el producto terminado.
- **Cantidad:** stock actual del producto o materia en el inventario de la empresa.
- **Unidad de medida:** cantidad estandarizada de compra o venta.
- **Estado venta:** un estado que indique si el producto aún se produce o ya no.

La Tabla 3.1 contiene datos de ejemplos proporcionados por la empresa, de algunas de las materias primas que utilizan en la producción de su cerveza:

#	Nombre Producto	Costo unit.	Stock	U. medida compra
1	Lúpulo Athanum	2.75	7.5	lb
2	Lúpulo Bravo	3.50	9	lb
3	Levadura ale	5.75	22	g
4	Cebada Malteada Tipo I	4.00	20	kg

TABLA 3.2 **DATOS EJEMPLO DE MATERIAS PRIMAS.**
(MORÁN JORQUERA, 2018).

La Tabla 3.2 por su lado, contiene datos de ejemplos proporcionados por la empresa de sus productos terminados:

#	Nombre Producto	Precio		U. medida venta	Estado
		unit.	Stock		
1	Raspberry Red Ale Barril	82.00	5	barriles	Se Produce
2	Raspberry Red Ale Botella	2.50	40	botellas	Se Produce
3	American Pale Ale Barril	75.00	2	barriles	Se Produce
4	Hopzilla IPA Barril	82.00	3	barriles	Se Produce
5	Milky Stout Botella	3.00	25	botellas	No se produce

TABLA 3.3 **DATOS EJEMPLO DE PRODUCTOS.**
(MORÁN JORQUERA, 2018).

Adicionalmente, la empresa adquiere la materia prima necesaria para la producción de su cerveza a distintos proveedores alrededor del país. Al contar en la actualidad con diversos proveedores, la empresa requiere manejar la información de estos de una manera eficiente. La información que requieren de cada uno de sus **proveedores** es la siguiente:

- **Cédula / RUC:** un número que identifique a cada proveedor.
- **Nombre:** nombre de la empresa o del proveedor.
- **Ciudad:** ciudad donde opera el proveedor.
- **Dirección:** dirección física del proveedor.
- **Teléfono:** un teléfono celular o fijo para contactar al proveedor.
- **Email:** una dirección de correo electrónico para contactar al proveedor.

La Tabla 3.3 contiene datos de ejemplos de algunos proveedores con la información requerida, aunque con datos no reales, ya que es información que la empresa se reserva:

Cédula / RUC	Nombre	Ciudad	Dirección	Telf. Fijo	Celular	Email
1001141629	Hop Dist.	Quito	Guaranda 217	2242206	0998	em1@gm.com
		Guayaqui				
1714381025	La Favorita	1	Av. 9 Octubre	2242207	0998	em2@gm.com
1715881767	Insu Brew	Quito	Av. Eloy Alfaro	2242208	0998	eml3@gm.com

TABLA 3.4 *DATOS EJEMPLO DE PROVEEDORES.*
(MORÁN JORQUERA, 2018).

Cada vez que la empresa compra materia prima a sus proveedores, los directivos requieren que se cree una orden de compra al proveedor. De esta manera, se podrá controlar de manera más eficiente el stock de productos que la empresa necesita y con los que cuenta actualmente. Es necesario que en cada orden de compra se registren todos los productos que se solicitaron al proveedor, y la cantidad exacta de los mismos.

La información que la empresa requiere en cada **orden de compra** sería la siguiente:

- **Número de orden:** un número que identifique a cada compra.
- **Proveedor:** un número que identifique al proveedor al que se realizó la compra.
- **Fecha:** fecha en la que se realizó la compra.
- **Productos:** lista de los productos que se compraron al proveedor.
- **Cantidad de los productos:** cantidad de cada producto que se solicitó.
- **Subtotal:** el total de dinero que se debe pagar antes de agregarle el IVA.
- **IVA:** el total de IVA o impuesto que se paga en la compra.
- **Total:** el total de dinero que se debe pagar al proveedor por la compra.

Con esta información, la empresa puede manejar su inventario. De esta manera, los usuarios podrán consultar y modificar de ser necesario toda la información de los productos que se encuentran en el inventario de la empresa.

En la Figura 3.1, tenemos un ejemplo de una orden de compra común en la empresa:

Orden de Compra para Saint Roots			No. 00028	
Proveedor:		Monkey Brewing		
Fecha:		30/5/2018		
#	Artículo	Cantidad	Costo	Subtotal
1	Lúpulo Amber Ale	5	2.5	12.5
2	Lúpulo Athanum	5	4	20
3	Levadura	10	5.75	57.5
Firma Autorizada			Subtotal	90
_____			IVA	10.8
			Total	100.8

FIGURA 3.9 **DATOS EJEMPLO DE ORDEN DE COMPRA.**
(MORÁN JORQUERA, 2018).

3.1.2. Producción de Cerveza

Los directivos de la empresa han tomado la decisión de escoger por calendario los días que serán destinados para elaborar lo que se denomina una orden de producción. Por cada orden de producción, se elabora un solo tipo de cerveza. La empresa requiere que se cree una orden de producción cada día que se empieza la producción de una cerveza. La empresa, para cada orden de producción, utiliza los insumos que se encuentran actualmente en el inventario.

De esta manera, cuando comienza el proceso de elaboración de la cerveza, el encargado reúne todos los insumos necesarios de bodega para la cerveza que va a elaborar. Una vez ha utilizado la cantidad de material que necesitaba para la producción, se vuelve a almacenar el resto en bodega. Finalmente, cuando ha finalizado todo el proceso de producción y se obtiene ya determinada cantidad de la cerveza elaborada, es necesario almacenar todo este producto en la bodega de la empresa.

Es decir, la empresa tiene los siguientes requerimientos para su proceso de producción:

1. Crear una orden de producción cada día que se empiece la elaboración de un tipo de cerveza artesanal.
2. Registrar la materia prima que se utiliza en cada orden de producción y la cantidad de cada una.
3. Registrar la cerveza (producto terminado) que se obtiene por cada orden de producción, y la cantidad que se obtuvo de la misma.

La información que la empresa requiere en cada **orden de producción** sería la siguiente:

- **Fecha inicio:** la fecha en que inicia la producción de determinada cerveza.
- **Fecha fin:** la fecha en que finaliza la producción de determinada cerveza.
- **Producto:** referencia al producto específico que se obtendrá en la producción.
- **Cantidad:** cantidad de cada producto que se obtiene, sea barriles o botellas.
- **Cantidad planificada:** cantidad que se planea obtener del producto.
- **Cantidad obtenida:** cantidad que se obtuvo realmente del producto.

Adicionalmente, para cada orden de producción, se requiere registrar la materia prima que se utilizó en la elaboración de la cerveza y la cantidad usada de cada una.

En la Figura 3.2, tenemos un ejemplo de una orden de producción común en la empresa donde se incluye toda la información necesaria del producto terminado:

Orden de Producción Saint Roots		No. 00017
Fecha inicio:	13/5/2018	
Fecha fin:	30/5/2018	
Cerveza producida:	Red Amber Ale	
Cantidad Planf.:	75 litros	
#	Producto	Cantidad
1	Red Amber barril	2
2	Red Amber botellas	25
Cantidad obtenida		73 litros

FIGURA 3.10 **DATOS EJEMPLO DE ORDEN DE PRODUCCIÓN.**
(MORÁN JORQUERA, 2018).

3.1.3. Distribución del producto

El encargado del departamento comercial de la empresa es quien mantiene contacto con los clientes de la empresa para realizar la distribución del producto requerido. Al ya tener dos años en el mercado, la cervecería actualmente cuenta con diversos clientes. Es por esto que la empresa requiere gestionar la información de cada cliente para asegurar una mayor eficiencia en la distribución de sus productos.

La información que la empresa requiere de sus **clientes** es la siguiente:

- **Cédula / RUC:** un número que identifique a cada cliente.
- **Nombre:** nombre de la empresa o del cliente.
- **Ciudad:** ciudad donde reside el cliente.
- **Dirección:** dirección del cliente para la entrega del producto.
- **Teléfono:** un teléfono celular o fijo para contactar al cliente.
- **Email:** una dirección de correo electrónico para contactar al cliente.

La Tabla 3.4 contiene datos de ejemplos de algunos clientes con la información requerida, aunque con datos no reales, ya que es información que la empresa se reserva:

Cédula /		Telf.				
RUC	Nombre	Ciudad	Dirección	Fijo	Celular	Email
1001141629	Wilson Moran	Quito	Guaranda 217	2242206	0998	email1@gm.com
1714381025	Verónica Jorq	o	Av. 9 Octubre	2242207	0998	email2@gm.com
1715881767	La Roots S.A	Quito	Eloy Alfaro	2242208	0998	email3@gm.com

TABLA 3.5 **DATOS EJEMPLO DE CLIENTES.**
(MORÁN JORQUERA, 2018).

La empresa requiere que cada vez que un cliente realiza un pedido, se cree una orden de pedido. De esta manera, se podrá controlar de manera más eficiente la distribución de los productos requeridos por el cliente. Es necesario que en cada orden de pedido se registren los productos que el cliente solicita, y la cantidad exacta de los mismos.

Una vez que se guarda la orden de pedido, se calcula el precio total de la misma y se informa de esto al cliente. La orden en este momento queda en un estado pendiente de entrega. La empresa requiere además que el encargado de distribución del producto pueda consultar y manejar la información de cada orden de pedido, de manera que pueda actualizar el estado de la misma una vez que se haya entregado.

La información que la empresa requiere para cada **orden de pedido** es la siguiente:

- **Número de orden:** un número que identifique a cada pedido.
- **Cliente:** un número que identifique al cliente que realizó el pedido.
- **Fecha:** fecha en la que se realizó el pedido.
- **Estado:** un código que identifique en qué estado se encuentra la orden, ya sea que está pendiente o ya haya sido entregada.
- **Productos:** lista de los productos que el cliente solicitó.
- **Cantidad de los productos:** cantidad de cada producto que el cliente solicitó.
- **Subtotal:** el total de dinero que se cobra antes de agregarle el IVA.

- **IVA:** el total de IVA o impuesto que se cobra en la venta.
- **Total:** el total de dinero que el cliente debe pagar por el pedido.

En la Figura 3.3, tenemos un ejemplo de una orden de pedido común de un cliente donde se incluye toda la información necesaria:

Orden de Pedido para Saint Roots		No. 000129		
Cliente:		La Roots S.A.		
Fecha:		30/5/2018		
#	Producto	Cantidad	Precio	Subtotal
1	IPA barriles	2	75	150
2	IPA botellas	25	2.5	62.5
3	Red ale botellas	10	3.5	35
	Firma Autorizada		Subtotal	247.5
	_____		IVA	29.7
			Total	277.2

FIGURA 3.11 **DATOS EJEMPLO DE UNA ORDEN DE PEDIDO.**
(MORÁN JORQUERA, 2018).

3.1.4. Gestión de Reporte Mensual

La empresa debe desembolsar una cantidad de dinero cada vez que realiza una orden de compra a sus proveedores. Este egreso tiene que ser registrado una vez que se ha cancelado el total de la orden de compra al proveedor. Por otro lado, la empresa recibe una cantidad de dinero cada vez que se entrega una orden de pedido a sus clientes. Este ingreso tiene que ser registrado de igual manera una vez que el cliente ha cancelado el total de la orden de pedido.

Es decir, la empresa requiere que se registre cada transacción, ya sea un ingreso generado por una orden de pedido del cliente, o un egreso generado por una orden de compra al proveedor. La empresa para esto, requiere generar un reporte del estado de

flujo de caja mensual, donde se reflejen las transacciones de cada mes. Este flujo de caja además de reflejar el total de ingresos y egresos del mes, debe indicar también el total del IVA que se pagó y que se cobró en el mes. De esta manera, el manejo de reportes del flujo de caja mensual se facilita mucho para la empresa, haciéndolo más eficiente.

La información que la empresa requiere para cada reporte es la siguiente:

- **Mes:** mes al que corresponde el determinado flujo de caja.
- **Año:** año el que corresponde el determinado flujo de caja.
- **Ingresos:** el total de ingresos que se registraron en el mes.
- **Egresos:** el total de egresos que se registraron en el mes.
- **Total:** la diferencia entre el total de ingresos y egresos del mes.
- **IVA cobrado:** el total del IVA que se cobró en el mes.
- **IVA pagado:** el total del IVA que se pagó en el mes.
- **IVA total:** la diferencia entre el total del IVA cobrado y el IVA pagado en el mes.

Lo que los directivos en conclusión requieren es poder generar mensualmente un reporte que refleje el total de ingresos vs el total de egresos del mes. Este flujo de caja generado brinda toda la información necesaria con respecto a los movimientos de efectivo de la organización.

3.2. Definición de Requerimientos

Una vez se ha hecho el respectivo levantamiento de información en las instalaciones de la cervecería y se han identificado los procesos, se logró coleccionar toda la información referente a las necesidades del usuario. De esta manera, se definen los siguientes requerimientos para el prototipo funcional que se va a desarrollar para la empresa.

3.2.1. Requerimientos Funcionales

Se han definido los siguientes requerimientos funcionales para la aplicación:

- F1.** El sistema será capaz de administrar clientes.
- F2.** El sistema será capaz de administrar proveedores.
- F3.** El sistema será capaz de administrar productos.

- F4.** El sistema será capaz de administrar las materias primas.
- F5.** El sistema será capaz de gestionar órdenes de pedido del cliente.
- F6.** El sistema será capaz de gestionar órdenes de compra al proveedor.
- F7.** El sistema será capaz de gestionar órdenes de producción de cerveza.
- F8.** El sistema será capaz de gestionar reportes de flujo de caja mensuales.

3.2.2. Requerimientos No Funcionales

Se han definido los siguientes requerimientos no funcionales para la aplicación:

1. El sistema en su totalidad será desarrollado y codificado en el lenguaje de programación JAVA.
2. El sistema será desarrollado en una arquitectura de 3 capas: Presentación (GUI), Dominio del Problema (DP), y Manejo de Datos (MD).
3. La información del sistema y los datos que maneja son almacenados en el motor de base de datos PostgreSQL.
4. El sistema brinda seguridad para el manejo de información, requiriendo una validación de usuario cada vez que se desea ingresar al mismo.

3.3. Modelo de Datos Conceptual

El diagrama conceptual de entidades y relaciones, puede ser dividido en distintos módulos para su mejor comprensión.

3.3.1. Módulo de Compras

El módulo de compras consiste en 3 entidades principales: Proveedor, Orden de Compra y Materia Prima. Al existir una relación de muchos a muchos entre la entidad Orden_Compra y Materia_Prima, se crea una nueva entidad “débil” que detallará qué materia prima, cuanta cantidad, y a qué costo se la compra en cada orden de compra.

En la Figura 3.4 se detalla el modelo conceptual del módulo de compras, con sus respectivas entidades, atributos y relaciones:

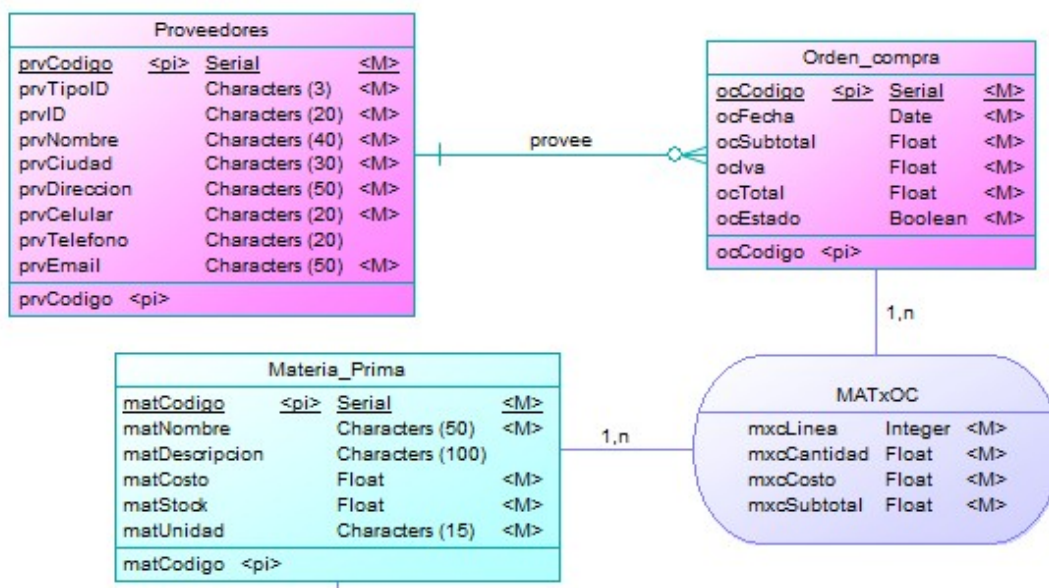


FIGURA 3.12 **MÓDULO DE COMPRAS.**
(MORÁN JORQUERA, 2018).

3.3.2. Módulo de Ventas

El módulo de ventas consiste en 3 entidades principales: Cliente, Orden de Pedido y Producto (hace referencia a la cerveza que produce la empresa). Al existir una relación de muchos a muchos entre la entidad Orden_Pedido y Producto, se crea una nueva entidad “débil” que detallará qué producto, cuanta cantidad, y a qué precio se lo vende en cada orden de pedido. En la Figura 3.5 se detalla el modelo conceptual del módulo de ventas, con sus respectivas entidades, atributos y relaciones:

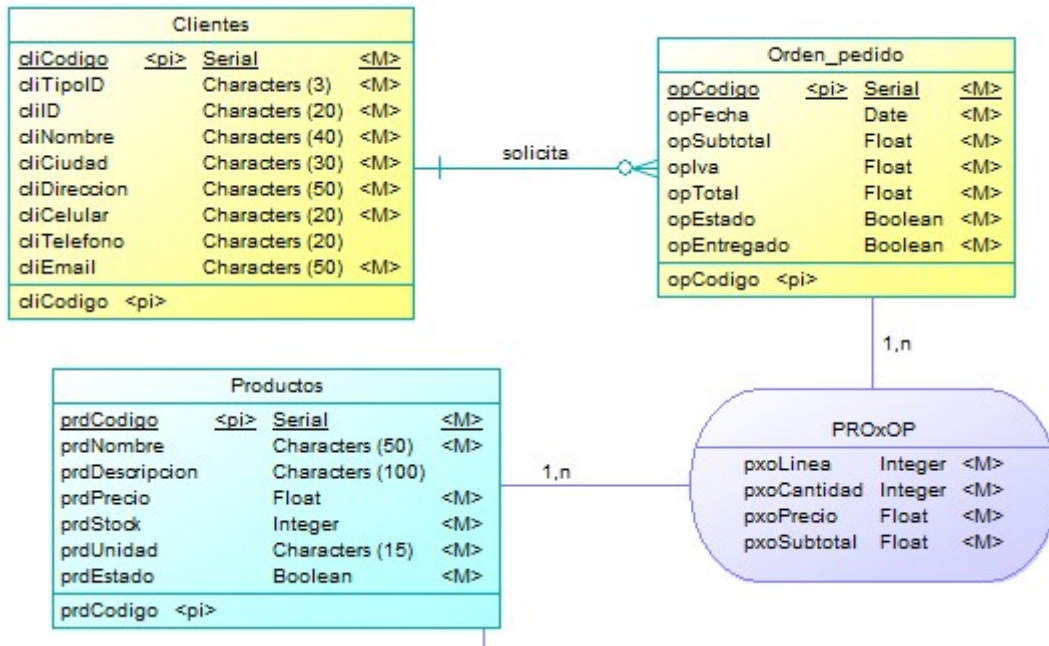


FIGURA 3.13 **MÓDULO DE VENTAS.**
(MORÁN JORQUERA, 2018).

3.3.3. Módulo de Producción

El módulo de producción consiste en una entidad principal llamada Orden de Producción, la misma que se relaciona con las entidades Producto y Materia Prima. Al existir una relación de muchos a muchos entre la entidad Orden_Produccion y Producto, se crea una nueva entidad “débil” que detallará qué producto y cuanta cantidad del mismo se obtuvo en la orden de producción. De igual manera, existe una relación de muchos a muchos entre la entidad Orden_Produccion y Materia_Prima, por lo que se crea una nueva entidad “débil” que detallará qué materia prima y cuanta cantidad de la misma se utilizó en la orden de producción.

En la Figura 3.6 se detalla el modelo conceptual del módulo de producción, con sus respectivas entidades, atributos y relaciones:

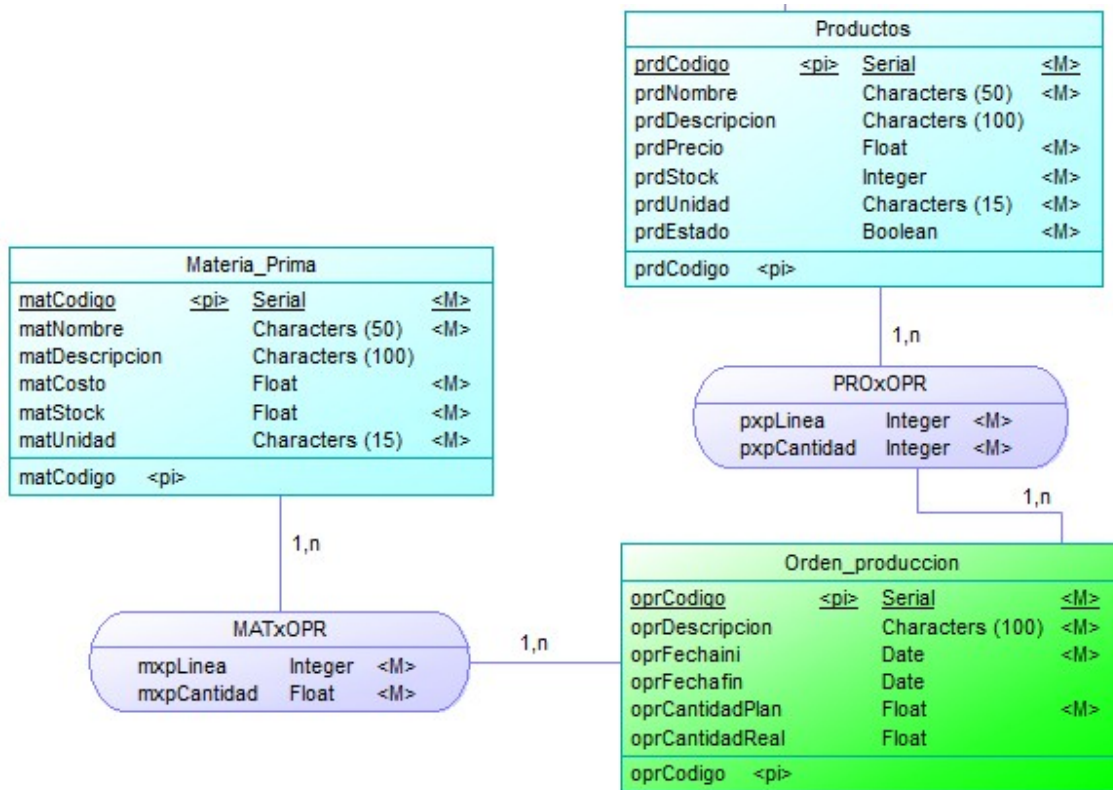


FIGURA 3.14 **MÓDULO DE PRODUCCIÓN.**
(MORÁN JORQUERA, 2018).

El modelo conceptual de datos completo con el que se trabajará para el desarrollo del sistema se muestra en la Figura 3.7. El modelo lógico de datos, por su lado, puede ser consultado en los anexos.

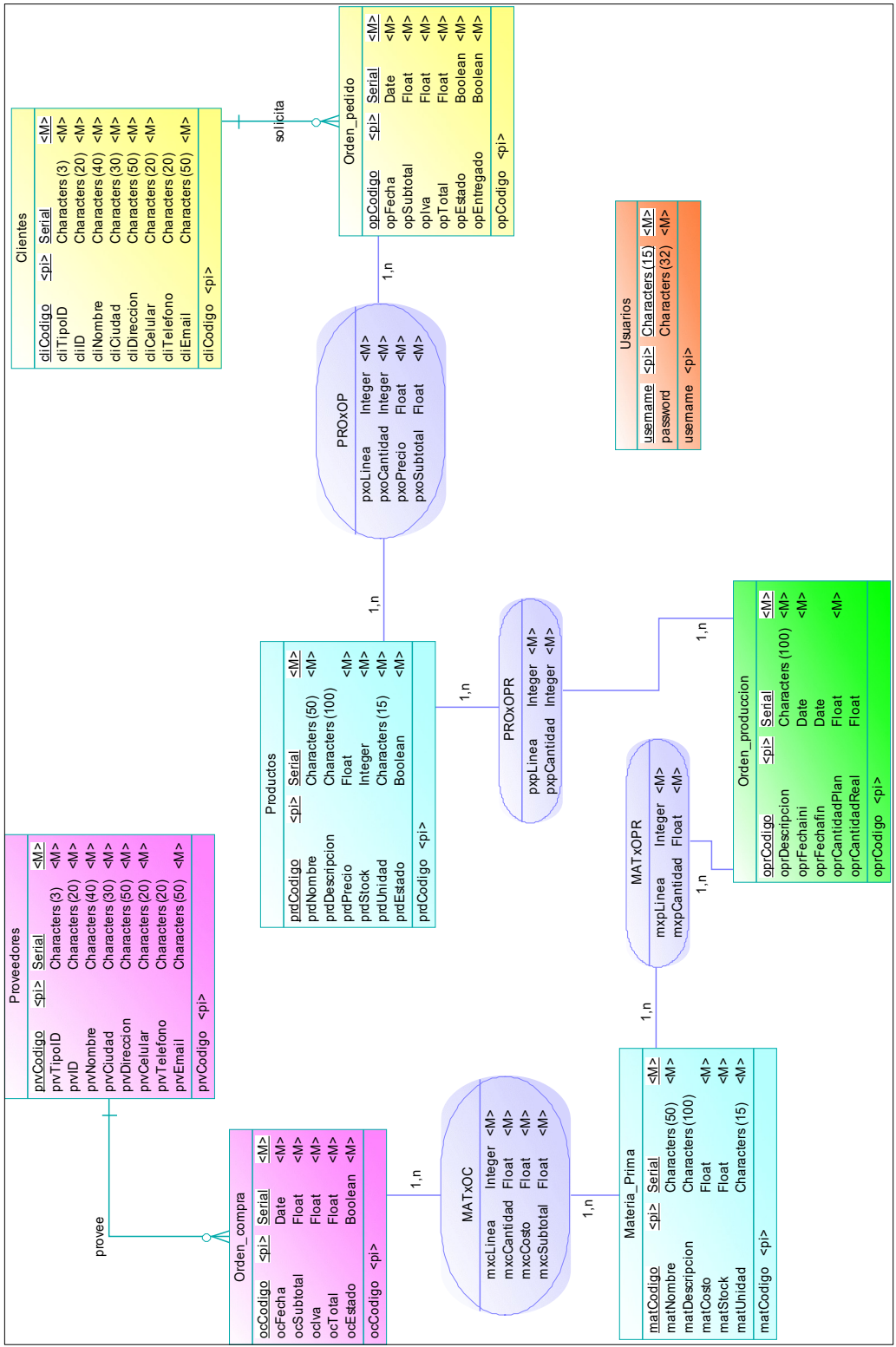


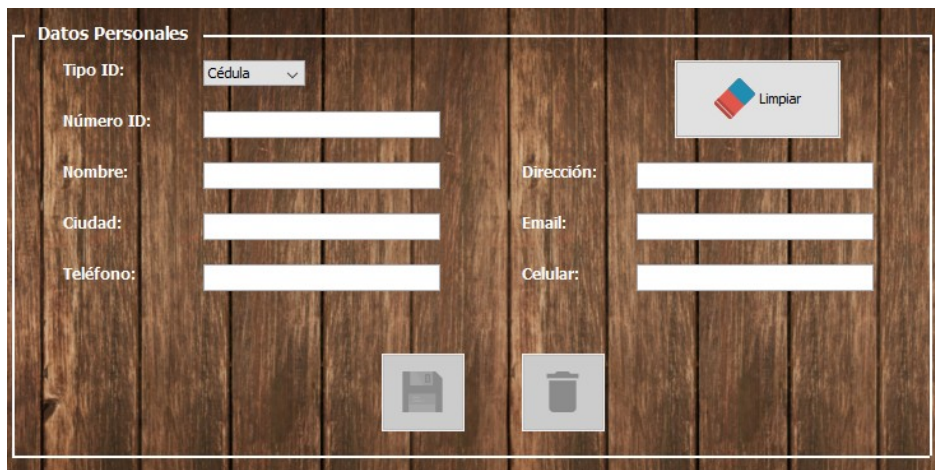
FIGURA 3.15 **MODELO CONCEPTUAL DE DATOS.**
(MORÁN JORQUERA, 2018).

3.4. Diseño de Interfaces

Se realizó un diseño de interfaz diferente para cada uno de los principales tipos de procesos a ejecutarse en el sistema, de los cuales podemos destacar los siguientes:

3.4.1. Interfaz para CRUD básico

Esta interfaz servirá para el ingreso de datos en cada una de las entidades principales, como son la de Clientes, Proveedores, Producto y Materia Prima. Existirá un campo para el ingreso de cada atributo, acompañado de dos botones: uno para guardar la información en la base y otro para borrar la información de la base. Ver Figura 3.8.



The image shows a web form titled "Datos Personales" set against a dark wood-grain background. The form contains the following elements:

- Tipo ID:** A dropdown menu with "Cédula" selected.
- Número ID:** A text input field.
- Nombre:** A text input field.
- Ciudad:** A text input field.
- Teléfono:** A text input field.
- Dirección:** A text input field.
- Email:** A text input field.
- Celular:** A text input field.
- Limpiar:** A button with a trash icon and the text "Limpiar".
- Guardar:** A button with a floppy disk icon.
- Borrar:** A button with a trash can icon.

FIGURA 3.16 **INTERFAZ CRUD.**
(MORÁN JORQUERA, 2018).

3.4.2. Interfaz para Consulta

Esta interfaz servirá para la consulta de datos en cada una de las entidades principales, como son la de Clientes, Proveedores, Producto y Materia Prima. Existirá un campo para ingresar el parámetro por el cual se desea buscar un objeto, acompañado de algunos botones: uno para realizar la búsqueda, otro para refrescar la información de la base, y el último sirve para limpiar la tabla. Ver Figura 3.9 para mayor detalle.

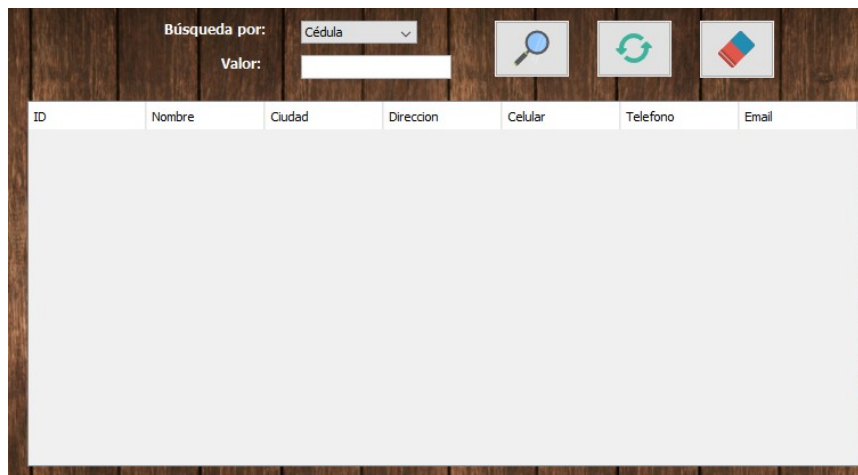


FIGURA 3.17 **INTERFAZ CONSULTA.**
(MORÁN JORQUERA, 2018).

3.4.3. Interfaz para Manejo de Orden

Esta interfaz servirá para el ingreso de datos de las entidades Orden de Pedido y Orden de Compra. Se podrá buscar un cliente o un proveedor respectivamente que esté almacenado en la base, para visualizar los datos del mismo, o crear uno nuevo de ser necesario. Estos son los datos pertenecientes a la Cabecera. Luego, se podrá detallar en la tabla los productos de la base que se venden o la materia prima de la base que se compra en cada orden; los cuales son datos pertenecientes al Detalle. Ver Figura 3.10 para mejor comprensión de la interfaz.

SAINT ROOTS BREWING CO.
RUC: 1714381025001

ORDEN DE PEDIDO
No.

ID Cliente:
 Nombre:
 Ciudad:

Dirección:
 Celular:

Fecha:
 Email:

Linea	Producto ID	Detalle	Precio	Cantidad	SubTotal

SubTotal:
 IVA:
 Total:

FIGURA 3.18 **INTERFAZ ORDEN.**
(MORÁN JORQUERA, 2018).


3.4.4. Interfaz para Manejo de Producción

Esta interfaz servirá para el ingreso de datos de la entidad Orden de Producción, la cual es el principal proceso de la cervecería. Existirá un botón para crear una nueva orden, especificando los datos de la misma como son la cantidad de producto y la fecha. A continuación existen dos tablas, una para detallar la materia prima que se utilizó en la producción, y otra para detallar los productos obtenidos finalizada la producción.

Existe otro botón que brinda la posibilidad de manejar una orden que ya ha sido creada previamente, de manera que se pueda seguir detallando la materia prima utilizada o el producto que se obtuvo. Finalmente, existen dos botones, uno que permite Guardar la información de la orden y otro que brinda la posibilidad de eliminar una orden. Ver Figura 3.11 para mejor comprensión de la interfaz.

ORDEN DE PRODUCCION

No.



SAINT ROOTS BREWING CO.
RUC: 1714381025001

Descripción:

Cantidad Planificada:* (litros)

Cantidad Obtenida: (litros)

Fecha inicio:*

Fecha fin:

Nueva Orden

Manejar Orden

MATERIALES UTILIZADOS

Linea	ID Material	Material	Unidad	Cantidad

+
X
↻

PRODUCTOS OBTENIDOS

Linea	ID Producto	Producto	Unidad	Cantidad

+
X
↻

📄
🗑️

FIGURA 3.19 INTERFAZ PRODUCCIÓN.
(MORÁN JORQUERA, 2018).

3.4.5. Interfaz para Reportes

Esta interfaz servirá para poder visualizar el reporte de Ingresos vs Egresos de acuerdo al mes y la fecha seleccionados por el usuario. Existe un botón que permite buscar los ingresos y egresos del mes, detallándolos en cada una de las tablas respectivamente, además de mostrar el total del mes. Existe además un botón que permite calcular la diferencia de Ingresos y Egresos, generando el reporte final para el usuario. Ver Figura 3.12 para mejor comprensión.

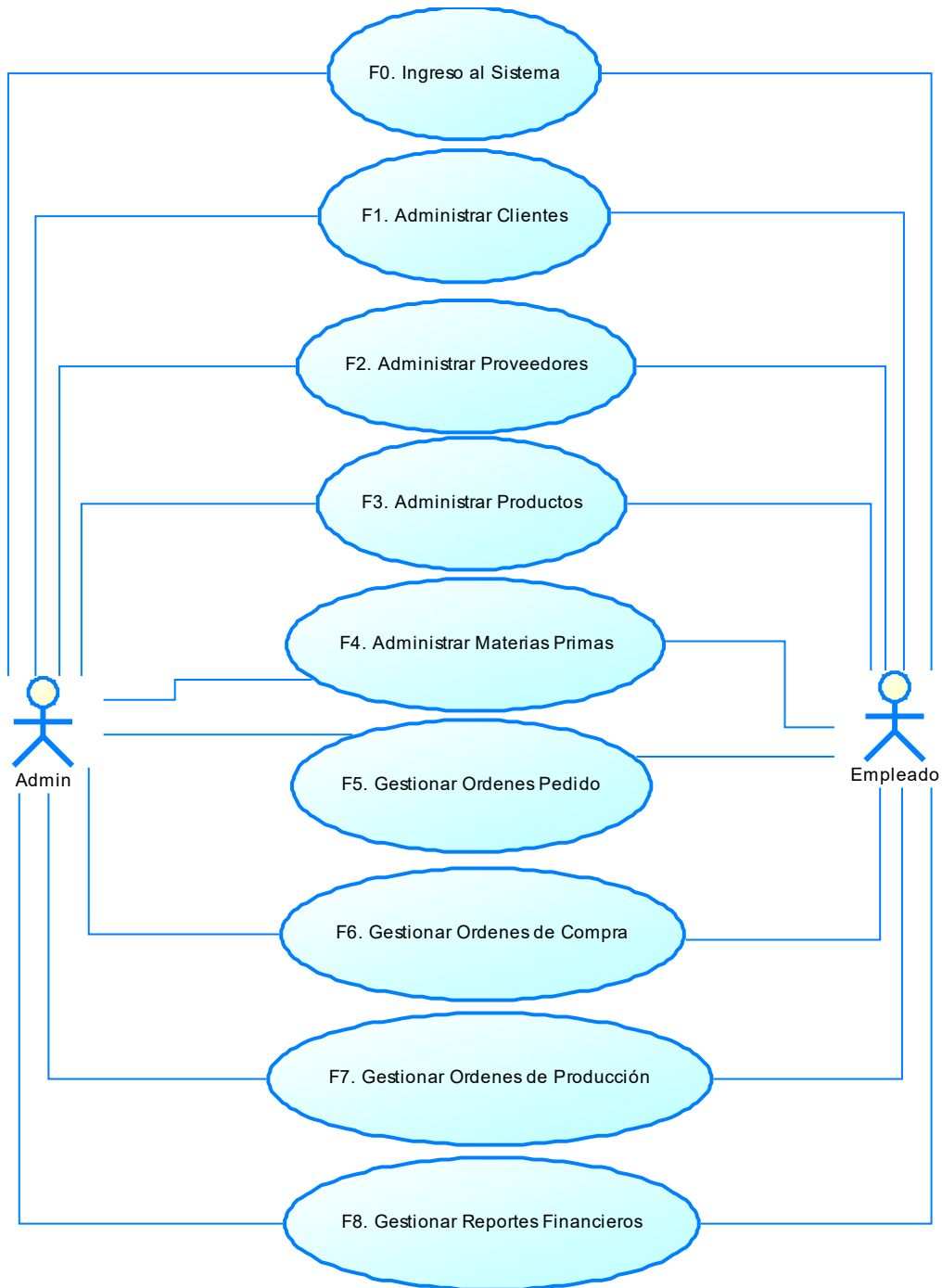


FIGURA 3.21 **DIAGRAMA GENERAL CASOS DE USO.**
(MORÁN JORQUERA, 2018).

A continuación se detallará en un nivel superior dos casos de usos específicos:

- F3. Administrar Productos (Ver Figura 3.14).
- F7. Gestionar Ordenes de Producción (Ver Figura 3.15).

El resto de los diagramas de casos de uso de siguiente nivel para cada una de las funcionalidades, pueden ser consultados en los anexos.

3.5.1. Siguiendo Nivel: Administración de Productos

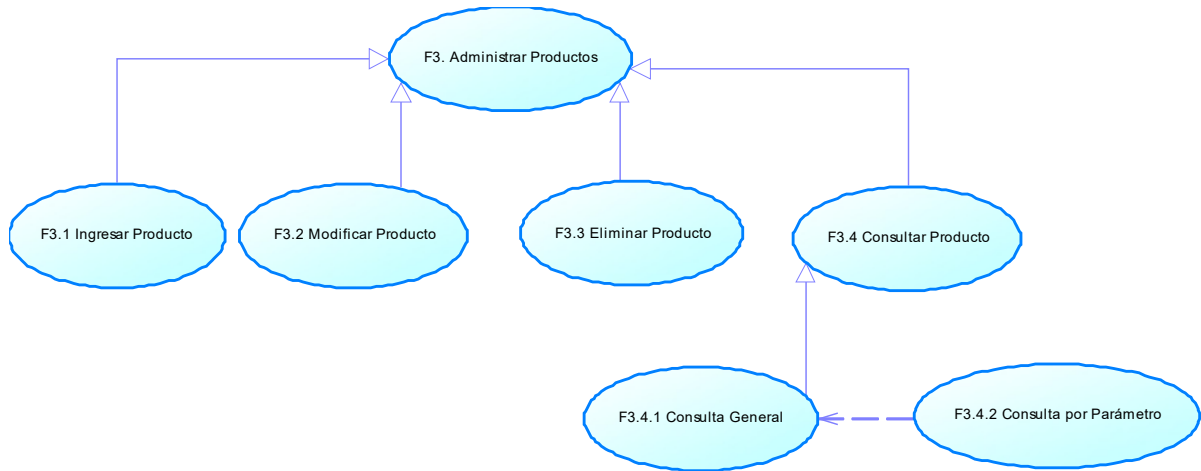


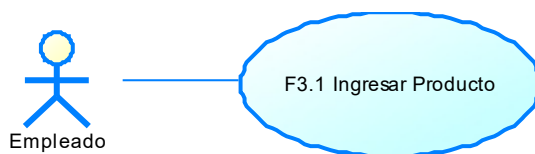
FIGURA 3.22 **CASO DE USO ADMINISTRAR PRODUCTOS.**
(MORÁN JORQUERA, 2018).

Caso de uso a Detalle:

1) ID: F3.1

Descripción: Se ingresa un nuevo producto terminado al sistema para poder manejar su información cuando sea necesario.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Inventarios”.
2. El sistema despliega el menú de Inventarios.
3. El actor presiona la opción “Productos”.
4. El sistema abre la ventana para administrar los productos.

5. El actor ingresa el nombre del producto.
6. El actor ingresa una descripción del producto.
7. El actor ingresa un precio para el producto.
8. El actor ingresa un stock inicial para el producto si corresponde.
9. El actor escoge la unidad de medida para el producto.
10. El actor escoge si el producto está activo o no.
11. El actor presiona el botón GUARDAR.
12. El sistema almacena los datos. **(E1)**

Flujo Alternativo:

5. Ver caso de Uso F3.4

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

2) ID: F3.2

Descripción: Permite modificar la información de los productos terminados que están en el sistema, tales como su descripción, su precio, etc.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Inventarios”.
2. El sistema despliega el menú de Inventarios.
3. El actor presiona la opción “Productos”.

4. El sistema abre la ventana para administrar los productos.
5. El actor escoge el producto que desea modificar.
6. El sistema despliega la información del producto seleccionado.
7. El actor modifica la información del producto seleccionado.
8. El actor presiona el botón GUARDAR.
9. El sistema modifica los datos. **(E1)**

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

3) ID: F3.3

Descripción: Permite eliminar un producto que está en el sistema cuando este ya no será de utilidad para la empresa.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Inventarios”.
2. El sistema despliega el menú de Inventarios.
3. El actor presiona la opción “Productos”.
4. El sistema abre la ventana para administrar los productos.
5. El actor escoge el producto que desea eliminar.
6. El sistema despliega la información del producto seleccionado.
7. El actor presiona el botón de Eliminar Producto.

8. El sistema elimina el producto del sistema. **(E1)**

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

4) **ID:** F3.4.1

Descripción: Permite consultar toda la información de todos los productos terminados que están registrados en el sistema de inventario.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Inventarios”.
2. El sistema despliega el menú de Inventarios.
3. El actor presiona la opción “Productos”.
4. El sistema abre la ventana para administrar los productos.
5. El actor presiona el botón de Refrescar para ver la información actualizada de los productos terminados.
6. El sistema verifica la información de los productos en el sistema. **(E1)**
7. El sistema despliega en la parte inferior la información de los productos.

Flujo Alternativo:

7. Ver caso de uso F3.1

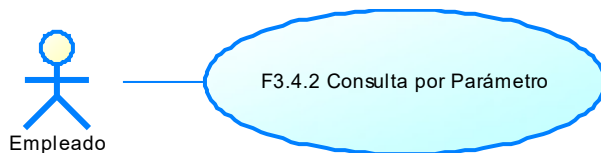
Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

5) ID: F3.4.2

Descripción: Permite consultar toda la información de un determinado producto registrado en el sistema por un parámetro específico.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Inventarios”.
2. El sistema despliega el menú de Inventarios.
3. El actor presiona la opción “Productos”.
4. El sistema abre la ventana para administrar los productos.
5. El actor selecciona el parámetro por el cual desea buscar el producto e ingresa la información del producto a buscar.
6. El actor presiona el botón “BUSCAR”.
7. El sistema busca la información del producto específico. **(E1)**
8. El sistema despliega en la parte inferior la información del producto.

Flujo Alternativo:

8. Ver caso de uso F3.1

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

3.5.2. Siguiete Nivel: Gestión Órdenes de Producción

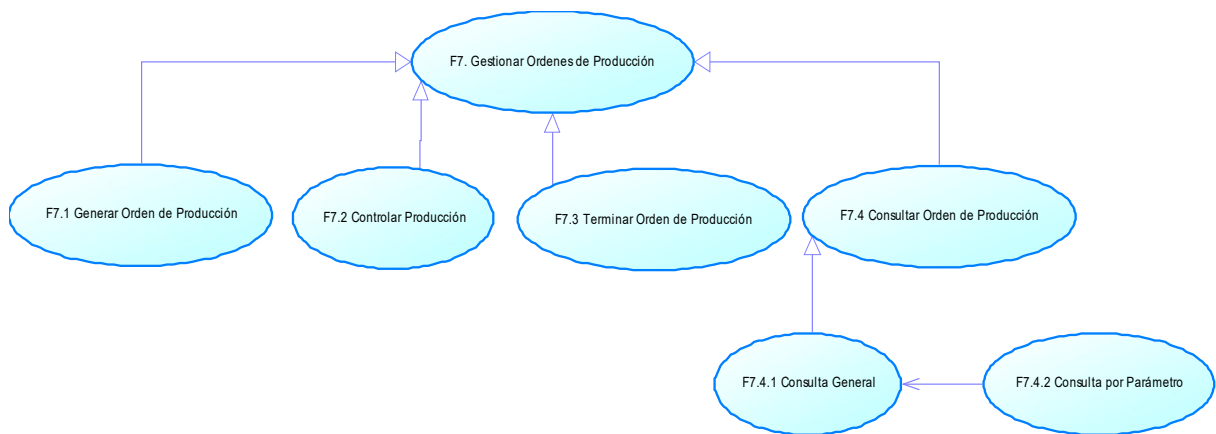


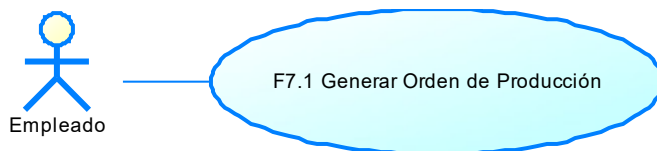
FIGURA 3.23 **CASO DE USO GESTIONAR ORDEN DE PRODUCCIÓN.**
(MORÁN JORQUERA, 2018).

Caso de uso a Detalle:

1) ID: F7.1

Descripción: Se genera una nueva orden de producción donde se detallan aspectos generales de la nueva orden de producción.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Producción”.
2. El sistema despliega el menú de Producción.

3. El actor presiona la opción “Orden de Producción”.
4. El sistema abre la ventana para gestionar las órdenes de producción.
5. El actor presiona el botón de Generar una Nueva Orden.
6. El sistema asigna un número identificador para la orden. **(E1)**
7. El actor ingresa una descripción de la orden de producción.
8. El actor ingresa la fecha de inicio de la nueva orden de producción.
9. El actor ingresa la cantidad planificada de cerveza a obtener en el proceso.
10. El actor presiona el botón GUARDAR.
11. El sistema almacena los datos. **(E1)**

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

2) ID: F7.2

Descripción: Se detalla la materia prima que se va utilizando en determinada orden de producción y la cantidad de la misma.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Producción”.
2. El sistema despliega el menú de Producción.
3. El actor presiona la opción “Orden de Producción”.
4. El sistema abre la ventana para gestionar las órdenes de producción.

5. El actor presiona el botón de Manejar una Orden.
6. El sistema abre la ventana de Búsqueda de Órdenes de Producción.
7. El actor ingresa el parámetro por el cual desea buscar la orden a manejar.
8. El sistema verifica las órdenes de producción que cumplan los parámetros. **(E1)**
9. El sistema despliega las órdenes de producción encontradas.
10. El actor selecciona la orden de producción que desea controlar.
11. El sistema despliega la información de la orden de producción.
12. El actor presiona el botón para añadir materia prima.
13. El sistema abre la ventana de Búsqueda de Materia Prima.
14. El actor ingresa el parámetro por el cual desea buscar la materia prima.
15. El sistema verifica las materias primas que cumplan los parámetros. **(E1)**
16. El sistema despliega las materias primas encontradas.
17. El actor selecciona la materia prima que utilizó.
18. El sistema despliega la información de la materia prima.
19. El actor ingresa la cantidad que se utilizó de la misma.
20. El actor presiona el botón GUARDAR.
21. El sistema almacena los datos. **(E1)**

Flujo Alternativo:

10. Ver caso de uso F7.1 o F7.4

17. Ver caso de uso F4.1 o F4.4

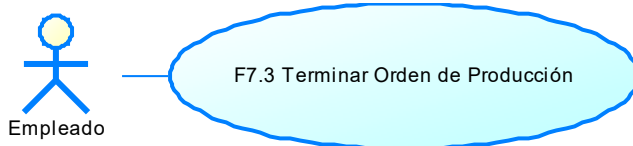
Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

3) ID: F7.3

Descripción: Se detalla el producto terminado que se obtuvo una vez finalizada una determinada orden de producción y la cantidad del mismo.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Producción”.
2. El sistema despliega el menú de Producción.
3. El actor presiona la opción “Orden de Producción”.
4. El sistema abre la ventana para gestionar las órdenes de producción.
5. El actor presiona el botón de Manejar una Orden.
6. El sistema abre la ventana de Búsqueda de Órdenes de Producción.
7. El actor ingresa el parámetro por el cual desea buscar la orden a terminar.
8. El sistema verifica las órdenes de producción que cumplan los parámetros. **(E1)**
9. El sistema despliega las órdenes de producción encontradas.
10. El actor selecciona la orden de producción que desea terminar.
11. El sistema despliega la información de la orden de producción.
12. El actor presiona el botón para añadir producto.
13. El sistema abre la ventana de Búsqueda de Producto.
14. El actor ingresa el parámetro por el cual desea buscar el producto terminado.
15. El sistema verifica los productos que cumplan los parámetros. **(E1)**
16. El sistema despliega los productos encontrados.
17. El actor selecciona el producto que se obtuvo en la orden.

- 18. El sistema despliega la información del producto.
- 19. El actor ingresa la cantidad que se obtuvo del mismo.
- 20. El actor presiona el botón GUARDAR.
- 21. El sistema almacena los datos. **(E1)**

Flujo Alternativo:

- 10. Ver caso de uso F7.1 o F7.4
- 17. Ver caso de uso F3.1 o F3.4

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

4) ID: F7.4.1

Descripción: Es posible consultar todas las órdenes de producción que han sido creadas hasta el momento.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Producción”.
2. El sistema despliega el menú de Producción.
3. El actor presiona la opción “Consulta de Orden de Producción”.
4. El sistema abre la ventana para consultar las órdenes de producción.
5. El actor presiona el botón de Mostrar Todas las órdenes.
6. El sistema verifica las ordenes de producción creadas **(E1)**.

7. El sistema despliega la información de todas las órdenes de producción.

Flujo Alternativo:

7. Ver caso de uso F7.1

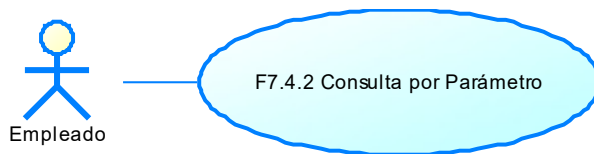
Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

5) ID: F7.4.2

Descripción: Es posible consultar las órdenes de producción que han sido creadas hasta el momento buscando por un parámetro determinado.

Autores: Empleado



Flujo Principal:

1. El actor presiona el módulo de “Producción”.
2. El sistema despliega el menú de Producción.
3. El actor presiona la opción “Consulta de Orden de Producción”.
4. El sistema abre la ventana para consultar las órdenes de producción.
5. El actor ingresa el parámetro por el cual desea buscar la orden.
6. El sistema verifica las órdenes de producción que cumplan los parámetros. **(E1)**
7. El sistema despliega las órdenes de producción encontradas.
8. El actor selecciona la orden de producción que desea consultar específicamente.
9. El sistema despliega la información de la orden de producción.

Flujo Alterno:

7. Ver caso de uso F7.1 o F7.4.1

9. Ver caso de uso F7.2 o F7.3

Excepciones:

Motivo	Solución
E1: No hay conexión con la BDD.	Llamar al administrador de la BDD.

3.6. Diagrama de Paquetes

El diagrama de paquetes es una herramienta útil en el diseño de cualquier sistema, que sirve para agrupar elementos estáticos. Cada paquete es una especie de caja que permite agrupar clases, objetos, casos de uso e incluso otros paquetes. Este diagrama permite además plantear la arquitectura del sistema a nivel macro.

La Figura 3.16 muestra el diagrama de paquetes del sistema desarrollado, donde se incluye el paquete de Interfaces de Usuario, de Dominio del Problema, y el de Manejo de Datos, además de un paquete auxiliar que contiene todas las imágenes e iconos con los que se desarrollaron las interfaces de usuario.

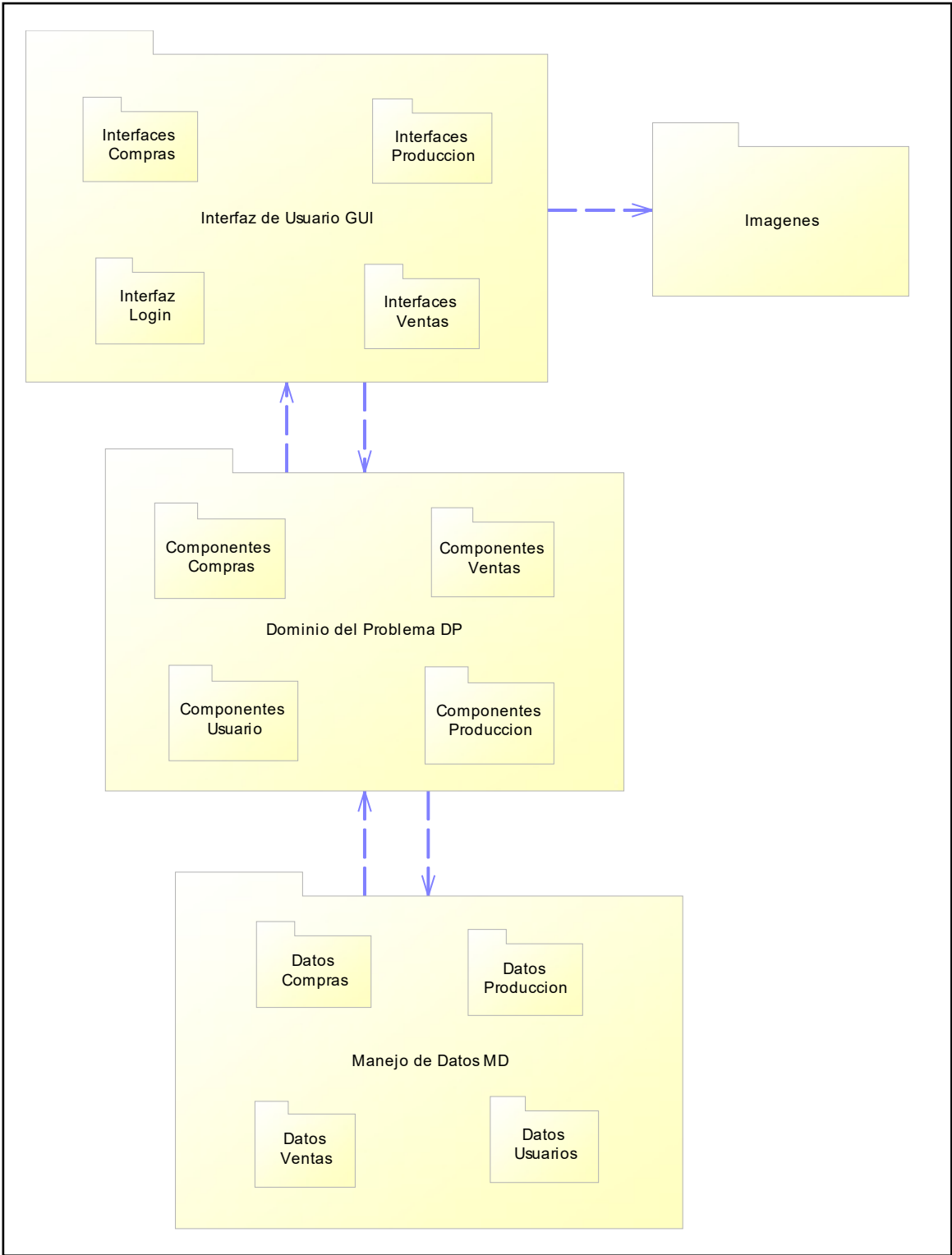


FIGURA 3.24 **DIAGRAMA DE PAQUETES.**
(MORÁN JORQUERA, 2018)

Capítulo 4 Desarrollo

Al utilizar una metodología ágil para el desarrollo del sistema, como lo es KanBan, se pudo obtener un producto final capaz de adaptarse de mejor manera a las necesidades del usuario. Se realizaron algunas iteraciones de desarrollo, y en cada una se fueron entregando pequeños prototipos evolutivos al cliente, añadiendo nuevas funcionalidades en cada una de estas.

Adicionalmente, en cada iteración se tomó muy en cuenta la retroalimentación del usuario, lo que permitió desarrollar el producto con un ciclo de vida incremental acorde a las principales necesidades del usuario. Esto permitió además, añadir pequeñas funcionalidades extra que el usuario encontró bastante útiles al momento de probar el sistema. Cada una de las iteraciones es descrita a continuación:

4.1. Iteración 01

4.1.1. Puntos Desarrollados

En esta primera iteración de desarrollo del prototipo funcional evolutivo, se desarrollaron las siguientes funcionalidades:

1) Conexión de Base de Datos al IDE

El script creado a partir del modelo de datos, fue implementando en el motor de base de datos PostgreSQL v10.4, creando una base de datos llamada *saintroots1*. El script de dicha base de datos puede ser consultado en los anexos. Una vez que la base fue creada, fue necesario instalar el driver JDBC v42.2.2 para PostgreSQL.

Finalmente, en el IDE NetBeans v8.2 se creó el proyecto con 3 distintos paquetes que reflejan las 3 capas de la arquitectura en la que se basará el sistema: GUI (Interfaz Gráfica, Presentación), DP (Dominio del Problema, Lógica del Negocio) y

MD (Manejo de Datos, Persistencia). Añadimos el driver JDBC descargado anteriormente y el ambiente de desarrollo está listo para comenzar con la codificación del sistema.

2) CRUD Clientes y Proveedores

Se crean 3 clases para cada una de las entidades, una en cada paquete. De esta manera, se asegura que la capa de Presentación se comuniquen solo con la capa de Lógica del Negocio y no directamente con la de Manejo de Datos, y viceversa.

Así, únicamente la capa de Negocio es la que es capaz de pasar mensajes, datos e información entre ambas capas.

- En el paquete GUI se creó: *MenuClientes.java* y *MenuProveedores.java*, con interfaces de ingreso y consulta de datos.
- En el paquete DP se creó: *Cliente.java* y *Proveedor.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos.
- En el paquete MD se creó: *ClienteMD.java* y *ProveedorMD.java*, con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda por ID tanto de clientes como de proveedores.

De esta manera quedó desarrollada la Inserción, Actualización, Eliminación y Consulta (CRUD) de información para las entidades de Clientes y Proveedores.

4.1.2. Tablero KanBan

Fase de Análisis

Lead Time de la Fase de Análisis y Diseño: 8 horas

Ítems completados (Tarjetas): 4 tareas

Los valores de KanBan para la primera iteración de desarrollo fueron los siguientes:

Fase de Desarrollo Iteración 01

WIP: 2 ítems

Lead Time: 10 horas

Ítems completados (Tarjetas): 2 tareas

En la Figura 4.1 se muestra el estado del Tablero KanBan en la primera iteración de desarrollo, donde se realizaron dos tareas.

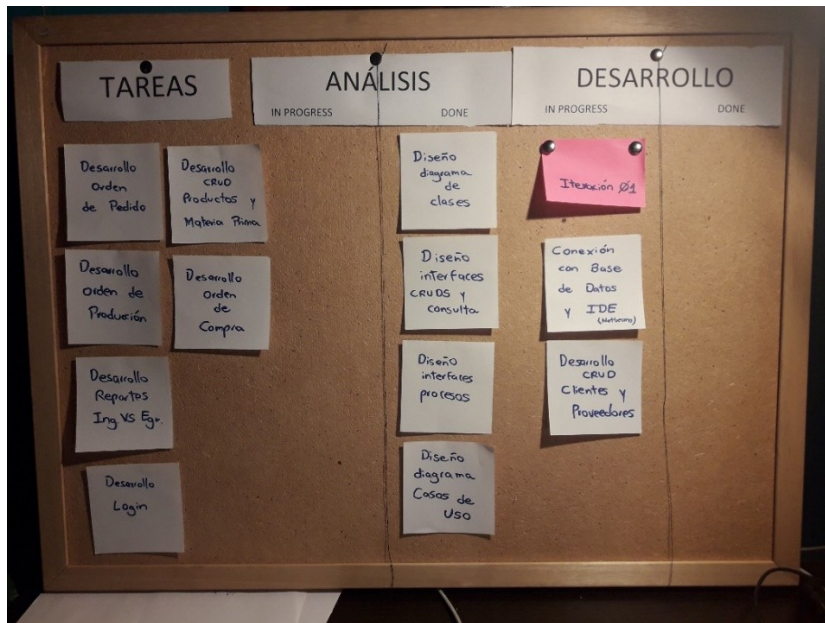


FIGURA 4.25 **TABLERO KANBAN 01.**
(MORÁN JORQUERA, 2018).

4.2. Iteración 02

4.2.1. Puntos Desarrollados

Al ser una metodología ágil, nos basamos en la entrega de pequeños prototipos evolutivos, añadiendo funcionalidades en cada una de las iteraciones. Es por eso que en esta segunda iteración de desarrollo se agregaron las siguientes funcionalidades:

1) CRUD Productos

- En el paquete GUI se creó: *MenuProductos.java*, con interfaces de ingreso y consulta de datos.
- En el paquete DP se creó: *Producto.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos.
- En el paquete MD se creó: *ProductoMD.java*, con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda tanto por el nombre del producto, como por la unidad de medida de los mismos (botellas y barriles).

2) CRUD Materia Prima

- En el paquete GUI se creó: *MenuMateria.java*, con interfaces de ingreso y consulta de datos.
- En el paquete DP se creó: *MateriaPrima.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos.
- En el paquete MD se creó: *MateriaMD.java*, con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda por nombre.

De esta manera quedó desarrollada la Inserción, Actualización, Eliminación y Consulta (CRUD) de información para las entidades de Productos y Materia Prima.

4.2.2. Tablero KanBan

Los valores de KanBan para la segunda iteración de desarrollo fueron los siguientes:

Fase de Desarrollo Iteración 02

WIP: 2 ítems

Lead Time: 8 horas

Ítems completados (Tarjetas): 1 tarea

En la Figura 4.2 se muestra el estado del Tablero KanBan en la segunda iteración de desarrollo, donde se realizó una tarea.



FIGURA 4.26 **TABLERO KANBAN 02.**
(MORÁN JORQUERA, 2018).

4.3. Iteración 03

4.3.1. Puntos Desarrollados

En esta iteración, al tener ya creados los CRUD de las entidades principales, se desarrollaron las clases necesarias para manejar la información tanto de las Órdenes de Compra como de las Órdenes de Pedido:

1) Orden de Compra

- En el paquete GUI se creó: *MenuCompras.java*, con interfaces de ingreso y consulta de datos. Se crearon además las clases *BuscarProveedor.java* y *BuscarMateria.java*, para poder añadir un proveedor y materia prima, respectivamente, desde la base de datos a la orden. Se creó también la clase *DetalleCompra.java*, que permite ver el detalle de una orden de compra.
- En el paquete DP se creó: *OrdenCompra.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos.

Para eso se creó además una clase llamada *MatxOc.java*, que guarda la información de la materia prima que se compra en cada orden.

- En el paquete MD se creó: *OCompraMD.java* con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda de orden de compra por parámetros tales como el ID del proveedor o el número de orden. Se creó además la clase *MatxOcMD.java* que permite consultar la información de la materia prima que se compra en cada orden guardada.

2) Orden de Pedido

- En el paquete GUI se creó: *MenuVentas.java*, con interfaces de ingreso y consulta de datos. Se crearon además las clases *BuscarCliente.java* y *BuscarProducto.java*, para poder añadir un cliente y productos, respectivamente, desde la base de datos a la orden. Se creó también la clase *DetalleVenta.java*, que permite ver el detalle de una orden de pedido.

- En el paquete DP se creó: *OrdenPedido.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos. Para eso se creó además una clase llamada *ProxOp.java*, que guarda la información de los productos que el cliente solicita en cada orden.
- En el paquete MD se creó: *OPedidoMD.java* con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda de orden de pedido por parámetros tales como el ID del cliente o el número de orden. Se creó además la clase *ProxOpMD.java* que permite consultar la información de los productos que se venden en cada orden guardada.

4.3.2. Métricas KanBan

Los valores de KanBan para la tercera iteración de desarrollo fueron los siguientes:

Fase de Desarrollo Iteración 03

WIP: 2 ítems

Lead Time: 9 horas

Ítems completados (Tarjetas): 2 tareas

En la Figura 4.3 se muestra el estado del Tablero KanBan en la tercera iteración de desarrollo, donde se realizaron dos tareas.

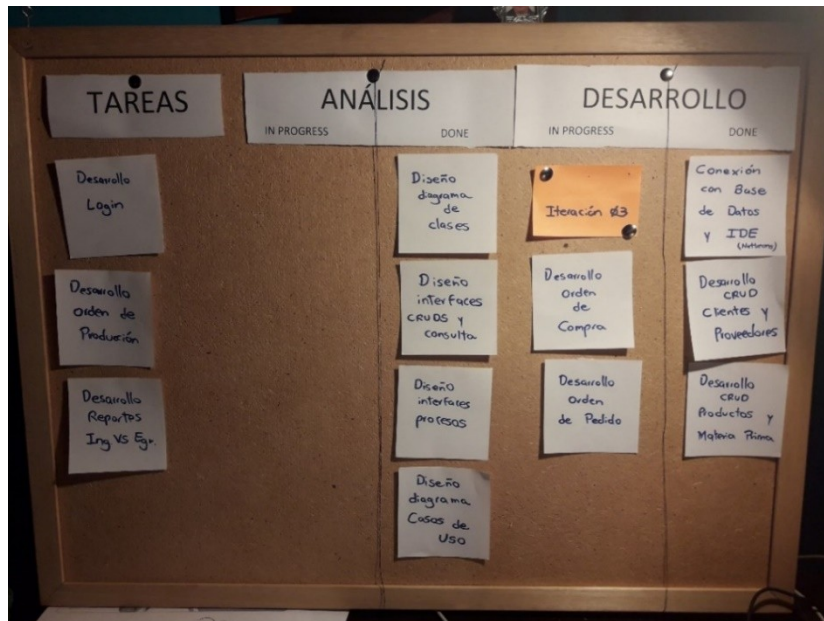


FIGURA 4.27 **TABLERO KANBAN 03.**
(MORÁN JORQUERA, 2018).

4.4. Iteración 04

4.4.1. Puntos Desarrollados

En esta iteración, al tener ya creados los procesos para manejo de información tanto de las Órdenes de Compra como de las Órdenes de Pedido, se desarrolló las clases necesarias para manejar la información relacionada a la producción de cerveza:

1) Orden de Producción

- En el paquete GUI se creó: *MenuProducción.java*, con interfaces de ingreso de datos. Esta interfaz permite ingresar la información básica de la orden, permite detallar la materia prima que se utilizó en la orden, y los productos que se obtuvieron una vez finalizada la producción. Se creó también la clase *DetalleProducción.java*, que permite ver el detalle de la orden.
- En el paquete DP se creó: *OrdenProducción.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos. Para esto se creó

además una clase llamada *MatxOpr.java* y *ProxOpr.java*, las cuales guardan la información de la materia prima que se utilizó en la orden y los productos que se obtuvieron una vez finalizada la orden.

- En el paquete MD se creó: *OProduccionMD.java* con funciones de acceso a la base de datos. Se incluyeron funciones de búsqueda de orden de producción por parámetros tales como el número de orden. Se creó además la clase *MatxOprMD.java* que permite consultar la información de la materia prima que se utilizó en cada orden, y la clase *ProxOpr.java* que permite consultar la información de los productos que se obtuvieron en cada orden.

4.4.2. Tablero KanBan

Los valores de KanBan para la cuarta iteración de desarrollo fueron los siguientes:

Fase de Desarrollo Iteración 04

WIP: 2 ítems

Lead Time: 8.5 horas

Ítems completados (Tarjetas): 1 tarea

En la Figura 4.4 se muestra el estado del Tablero KanBan en la cuarta iteración de desarrollo, donde se realizó una tarea.

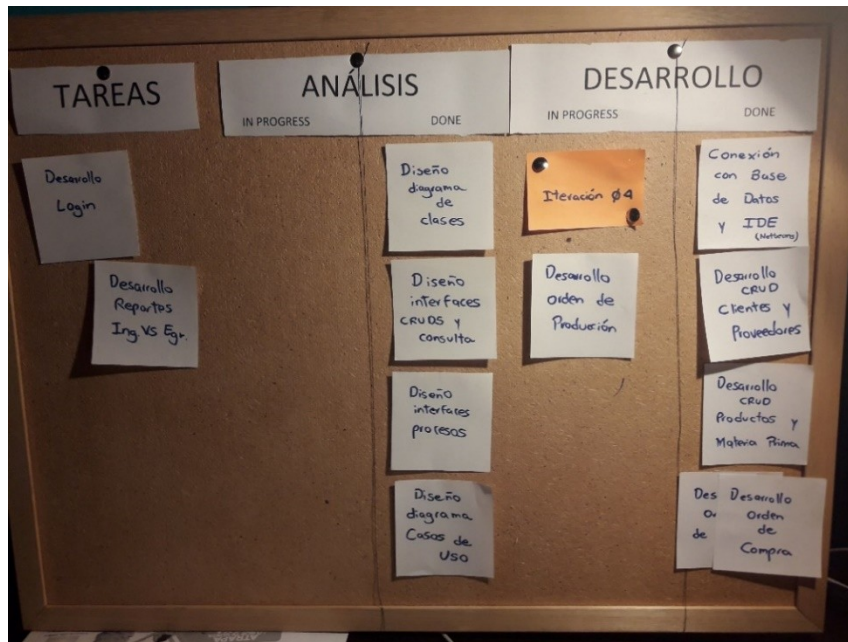


FIGURA 4.28 **TABLERO KANBAN 04.**
(MORÁN JORQUERA, 2018).

4.5. Iteración 05

4.5.1. Puntos Desarrollados

En esta iteración se creó el proceso final que permite generar reportes de Ingresos Vs Egresos según un mes y año seleccionados por el usuario. Se desarrolló además el formulario de Login, lo que permite que el usuario tenga que validar su identidad para poder hacer uso del sistema.

1) Reportes Ingresos vs Egresos

- En el paquete GUI se creó: *MenuReportes.java*, con interfaces de consulta de datos. Esta interfaz permite visualizar la información de los ingresos y egresos del mes seleccionado por el usuario.
- En el paquete MD se modificó la clase: *OCompraMD.java* y *OPedidoMD.java*, añadiendo funciones que permiten la búsqueda de información por parámetro, según un mes y un año determinados.

2) Formulario de Login

- En el paquete GUI se creó: *Validación.java*, con interfaz de ingreso de datos como el nombre de usuario y la contraseña.
- En el paquete DP se creó: *Usuario.java*, con métodos Getter y Setter para la obtención y seteo de datos, además de funciones que permitan pasar esta información a la capa de Manejo de Datos.
- En el paquete MD se creó: *UsuarioMD.java* con funciones de acceso a la base de datos. Se incluyeron funciones de validación de usuario, además de una función que permite encriptar la contraseña en MD5, para mayor seguridad.

4.5.2. Tablero KanBan

Los valores de KanBan para la quinta iteración de desarrollo fueron los siguientes:

Fase de Desarrollo Iteración 05

WIP: 2 ítems

Lead Time: 7 horas

Ítems completados (Tarjetas): 2 tareas

En la Figura 4.5 se muestra el estado del Tablero KanBan en la quinta iteración de desarrollo, donde se realizaron dos tareas.

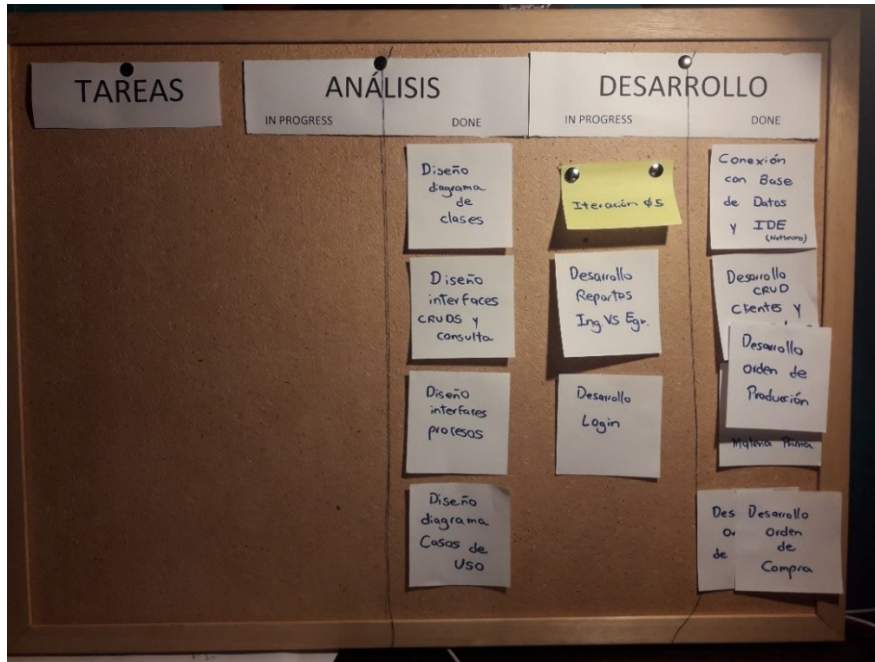


FIGURA 4.29 **TABLERO KANBAN 05.**
(MORÁN JORQUERA, 2018).

Capítulo 5 Pruebas

Las pruebas se realizan para establecer el grado de confianza en un sistema, la cual estará determinada por su grado de adherencia a las necesidades, requerimientos y procesos de negocio del usuario o del cliente. Es en base a estas pruebas en donde el usuario debe decidir si acepta o no el sistema que le está siendo entregado.

En este capítulo se detallan en primer lugar, las pruebas de integración que se ejecutaron con el cliente cada vez que se terminaba una iteración de desarrollo. Luego, una vez concluida la última iteración, se detallan los resultados de las pruebas de aceptación finales que se realizaron con el cliente.

5.1. Pruebas de Integración

En cada una de las pruebas de integración realizadas, se entregó un prototipo del sistema al cliente para que pueda probarlo. De esta manera, con sus observaciones y su valiosa retroalimentación, se realizaron los respectivos cambios al sistema para adaptarlo de mejor manera a sus necesidades.

5.1.1. Prueba 01

PRUEBAS DE INTEGRACIÓN 01	Fecha: 2018/04/20
Cliente	Saint Roots Brewing Co.
Cargo	Fabián Tenemaza - Gerente Comercial
Desarrollador	Martín Morán
Componentes Probados	Clientes, Proveedores

Con los componentes de Clientes y Proveedores desarrollados, se procedió a entregar el prototipo evolutivo al cliente para realizar las pruebas junto a él y obtener así su retroalimentación. La Tabla 5.1 muestra los resultados que se obtuvieron en las pruebas:

Observaciones	Cambios
En la consulta de información tanto de clientes como de proveedores, el cliente sugirió que sería útil poder buscarlos no solo por su número de ID, sino también por ciudad.	- Se aumentó la función de búsqueda por ciudad en la capa de Manejo de Datos. - Se aumentó la opción de buscar por ciudad en la interfaz gráfica de búsqueda.

TABLA 5.6 **PRUEBAS DE INTEGRACIÓN 01**
(MORÁN JORQUERA, 2018)

5.1.2. Prueba 02

PRUEBAS DE INTEGRACIÓN 02	Fecha: 2018/05/04
Cliente	Saint Roots Brewing Co.
Cargo	Fabián Tenemaza - Gerente Comercial
Desarrollador	Martín Morán
Componentes Probados	Productos, Materia Prima

Con los componentes de Productos y Materia Prima funcionando, se procedió a entregar el prototipo evolutivo al cliente para obtener retroalimentación del mismo. La Tabla 5.2 muestra los resultados de las pruebas:

Observaciones	Cambios
----------------------	----------------

Para la edición y actualización de la información de determinado producto o material, el cliente sugirió que sería mejor tener la posibilidad de escoger el producto a cambiar desde una tabla, ya que buscarlos por el código o el ID tal como se realizaba en las interfaces de Clientes y de Proveedores, no sería factible para los productos.

- Se juntó la interfaz de ingreso de datos y consulta, aumentando una tabla en la interfaz de ingreso. De esta manera, cuando se busque el producto cuya información se quiere actualizar, se pueda escogerlo y sus datos se carguen directamente en los campos destinados a la inserción de datos, brindando facilidad al usuario.

TABLA 5.7 **PRUEBAS DE INTEGRACIÓN 02**
(MORÁN JORQUERA, 2018)

5.1.3. Prueba 03

PRUEBAS DE INTEGRACIÓN 03	Fecha: 2018/05/18
Cliente	Saint Roots Brewing Co.
Cargo	Javier Morán - Gerente de Operaciones
Desarrollador	Martín Morán
Componentes Probados	Clientes, Proveedores, Productos, Materia Prima, Orden de Compra, Orden de Pedido

Con los componentes de Orden de compra y Orden de pedido desarrollados, se procedió a probar la integración de los mismos con los componentes desarrollados en las iteraciones anteriores.

La retroalimentación del usuario en esta iteración fue de gran ayuda ya que permitió corregir algunos errores, además de añadir funcionalidades importantes. La Tabla 5.3 muestra los resultados de estas pruebas:

Observaciones	Cambios
El cliente indicó que era necesario poder buscar las órdenes, tanto de compra como de pedido, no solo por el número de orden, sino también por la fecha en la que se realizó.	<ul style="list-style-type: none"> - Se añadió la opción de búsqueda por fecha en la interfaz de consulta de orden de compra y orden de pedido. - Se añadió una función de búsqueda por parámetro en la capa de Manejo de Datos de órdenes de pedido y órdenes de compra. - Se añadió la opción de cambiar el estado de una orden tanto en la interfaz de detalle de compras como en la de detalle de ventas.
El cliente indicó que era necesario poder modificar el estado de las órdenes, para poder anular aquellas que ya no son válidas.	<ul style="list-style-type: none"> - Se añadió una función de actualización de información en la capa de Manejo de Datos.
El cliente indicó que, exclusivamente para el caso de las órdenes de pedido, era necesario poder indicar si la orden ya fue entregada, o todavía está pendiente.	<ul style="list-style-type: none"> - Se añadió la opción de cambiar el estado de entregado en la interfaz de detalle de una orden de pedido. - Se añadió una función de actualización de información en la capa de Manejo de Datos.

TABLA 5.8 **PRUEBAS DE INTEGRACIÓN 03**
(MORÁN JORQUERA, 2018)

5.1.4. Prueba 04

PRUEBAS DE ITERACIÓN 04	Fecha: 2018/06/08
Cliente	Saint Roots Brewing Co.
Cargo	Javier Morán - Gerente de Operaciones
Desarrollador	Martín Morán
Componentes Probados	Productos, Materia Prima, Orden de Producción

Con el componente de Órdenes de producción desarrollado, se probó su integración con los componentes de productos y materia prima. La retroalimentación del usuario en esta iteración permitió añadir funcionalidades importantes al prototipo evolutivo. La Tabla 5.4 muestra los resultados de estas pruebas:

Observaciones	Cambios
El cliente indicó que era necesario poder buscar las órdenes no solo por el número de orden, sino también por la fecha de inicio de producción.	<ul style="list-style-type: none"> - Se añadió la opción de búsqueda por fecha en la interfaz de consulta de orden. - Se añadió una función de búsqueda por parámetro en la capa de Manejo de Datos.
El cliente sugirió la posibilidad de eliminar una orden, en caso de que haya sido creada por accidente y no tenga productos o materia prima asociados.	<ul style="list-style-type: none"> - Se añadió un botón para eliminar una orden seleccionada. - Se añadió una función de eliminación de datos en la capa de Manejo de Datos.

TABLA 5.9 **PRUEBAS DE INTEGRACIÓN 04**
(MORÁN JORQUERA, 2018)

5.1.5. Prueba 05

PRUEBAS DE INTEGRACIÓN 05	Fecha: 2018/06/22
Cliente	Saint Roots Brewing Co.
Cargo	Fabián Tenemaza - Gerente Comercial
Desarrollador	Martín Morán
Componentes Probados	Pedidos y Compras (Reportes), Usuario

Con los componentes de órdenes de compra y pedidos desarrollados, se probó la integración de estos con el módulo de reportes de ingresos vs egresos. Además, se probó la integración del módulo de usuarios con el resto del sistema.

En cada iteración se han ido añadiendo funcionalidades, considerando siempre la retroalimentación del usuario como lo más importante ya que así hemos desarrollado un prototipo evolutivo que se ajusta de mejor manera a las necesidades del cliente. La Tabla 5.5 muestra los resultados de estas pruebas:

Observaciones	Cambios
El cliente sugirió añadir una funcionalidad de cerrar sesión, de forma que solo se tenga que volver a acceder al mismo sin la necesidad de cerrarlo y volver a ejecutarlo.	Se añadió un menú que permite cerrar la sesión del usuario que había ingresado, volviendo de esta manera a la pantalla de Login.

TABLA 5.10 **PRUEBAS DE INTEGRACIÓN 05**
(MORÁN JORQUERA, 2018)

5.2. Pruebas de Aceptación

Una vez finalizadas todas las iteraciones de desarrollo del sistema, se procedió a realizar una prueba de aceptación final con el cliente, siendo los usuarios finales los directivos de la empresa; de tal manera que estos fueran capaces de probar el sistema evolutivo una vez que ya se han integrado todas las funcionalidades requeridas del mismo.

PRUEBAS DE ACEPTACIÓN	Fecha: 2018/07/02
Cliente	Saint Roots Brewing Co.
Cargo	Fabián Tenemaza - Gerente Comercial Javier Morán – Gerente de Operaciones
Desarrollador	Martín Morán

5.2.1. Resultados

El sistema de información que cuenta con todas las funcionalidades requeridas, generó algunas observaciones de los usuarios finales, que de manera general, quedaron satisfechos con el producto, tomando en cuenta que este no ha sido implementado aún en la empresa. La Tabla 5.6 muestra los resultados de las pruebas:

Observaciones	Cambios
El cliente sugirió cambiar algunos iconos de los botones, de forma que sea más amigable con el usuario y se adapte más al tema cervecero.	- Se cambiaron iconos asociados a diferentes botones, con temática de cerveza.
El cliente sugirió realizar cambios a la interfaz de Login, añadiendo imágenes que se adapten más a sus gustos.	- Se cambiaron algunas imágenes relacionadas a la interfaz de Login.

TABLA 5.11 **PRUEBAS DE ACEPTACIÓN**
(MORÁN JORQUERA, 2018)

5.2.2. Métricas KanBan

Fase de Pruebas Final

WIP: 2 ítems

Lead Time: 2 horas

Ítems completados (Tarjetas): 1 tarea [Pruebas de Aceptación con el Cliente]

5.3. Revisión KanBan Board

Una vez finalizadas las fases de Análisis y Diseño, Desarrollo, y por último la de Pruebas, se ha completado el ciclo de desarrollo del sistema, en donde se obtuvo como resultado de la unión de varios prototipos evolutivos, el sistema de información deseado y probado por el cliente.

El rendimiento es una medida importante en la metodología KanBan, ya que da una idea del número de elementos de trabajo que el equipo o el desarrollador es capaz de producir en un determinado período de tiempo. Los valores de rendimiento que se obtuvieron para este ciclo son los siguientes:

Ciclo Completado

Tiempo de Ciclo (Cycle Time): 52.5 horas

Tareas realizadas (Tarjetas): 13

Rendimiento (Throughput): 0.25 tarjetas por hora / 4.04 horas por tarjeta

Capítulo 6 Conclusiones y Recomendaciones

Una vez finalizado el desarrollo del sistema, y éste ha sido aceptado por el usuario, es posible obtener algunas conclusiones con respecto a diversos puntos involucrados en el presente trabajo de disertación, entre los cuales podemos destacar temas como la metodología de desarrollo utilizada, y también las herramientas tecnológicas que se utilizaron como son el IDE, el gestor de base de datos, el lenguaje de programación, la arquitectura, entre otros.

Con la finalización del presente trabajo de disertación, es fundamental además elaborar conclusiones acerca de la carrera cursada y de la universidad. De esta manera, podemos obtener algunas conclusiones incluso de la sociedad como tal, involucrando no solo temas tecnológicos, sino temas sociales en los que este tipo de trabajos de disertación pueden llegar a influir.

Finalmente, a partir de estas conclusiones, es necesario definir algunas recomendaciones de manera que nuestro trabajo sea aún más beneficioso. Estas son acciones que se sugiere realizar para lograr obtener resultados más favorables. Se incluyen además puntos que se desean mejorar para futuros trabajos.

6.1. Conclusiones

Las conclusiones que se obtuvieron a partir del presente trabajo de disertación fueron las siguientes:

- PostgreSQL fue el sistema de gestión de base de datos utilizado, ya que se decidió trabajar desde un comienzo con el paradigma de programación orientado a objetos, y este sistema al manejar bases de datos relacionales y ser además orientado a objetos, ofreció muchas ventajas en el desarrollo del prototipo. Es

posible concluir que a pesar de que su velocidad de respuesta pueda parecer a veces deficiente en bases pequeñas, como fue el caso, esta velocidad se mantiene aunque se aumente el tamaño de la base, fenómeno que no sucede con otro DBMS que pueden llegar a volverse críticamente lentos.

- Se utilizó NetBeans como entorno integrado de desarrollo para el sistema de información, ya que el mismo está hecho principalmente para el lenguaje de programación JAVA y este fue el lenguaje de programación con el que se decidió realizar el sistema. Este IDE permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes llamados módulos, brindando la capacidad de que sean extendidas con facilidad. Con esto, a fin de cuentas, el desarrollar la aplicación utilizando NetBeans como IDE es beneficioso para una pequeña empresa, ya que seguramente aparecerán nuevos requerimientos en un futuro, los que pueden ser fácilmente asociados a nuevos módulos, y así el sistema no tenga mayor dificultad para ser extendido.
- El uso de software libre y además gratuito como lo son NetBeans y PostgreSQL, permiten a una pequeña o mediana empresa destinar sus recursos a temas más importantes como lo son la ampliación de su infraestructura o de su producción, sin que sus intentos de crecimiento se vean afectados o mermados por hacer frente al pago de grandes cantidades de dinero en licencias. Se puede concluir además que el desarrollo de una aplicación utilizando licencias de software libre permiten una independencia tecnológica para el desarrollador, teniendo en cuenta que se tiene acceso al código fuente y que las licencias de software libre permiten la instalación del software en tantas máquinas como el usuario desee.

- Las metodologías ágiles involucran procesos evolutivos, lo que permite que se tenga una rápida respuesta a los cambios. Esto benefició mucho al desarrollo del sistema de información, ya que no fue necesario esperar hasta el final para corregir fallos y realizar cambios en el sistema. Las entregas parciales que se hacían al cliente permitieron mejorar la optimización de recursos y del tiempo para labores de pruebas y control. De lo anterior, se puede concluir que la intervención activa del cliente en el proceso fue clave en este trabajo, ya que siempre aportó con ideas importantes y opiniones puntuales sobre los resultados que se le iban entregando en cada iteración, permitiendo obtener un prototipo resultado de la unión de pequeño prototipos previamente probados y aceptados por el usuario.
- La colaboración con el cliente fue fundamental para obtener el prototipo final. En el desarrollo ágil, el cliente se vuelve un miembro más del equipo, que se integra y colabora con el grupo de trabajo para obtener un producto de calidad, acorde a las necesidades del mismo. Esto provocó además mejorar la capacidad de respuesta a los cambios, ya que se tuvo como valores de gestión ágil la anticipación y la adaptación.
- La empresa requiere conocer el precio de venta de sus productos terminados, mientras que para la materia prima este no es un dato relevante ya que no revenderán la misma. Es por esto que fue necesario manejar el inventario de la empresa con dos clasificaciones de productos: la materia prima por un lado, y el producto terminado como tal que es la cerveza que producen.
- La empresa vende y distribuye su producto terminado en dos presentaciones, que son en barriles y en botellas. Cada presentación por lo tanto tiene un precio de

venta diferente y una unidad de medida diferente; es por esto que cada presentación de cada tipo de cerveza que la empresa produce, tuvo que ser manejado como un producto terminado individual con sus respectivos atributos.

- En cada orden de producción que la empresa genera, se produce un solo tipo de cerveza. Sin embargo, en cada orden se pueden obtener distintos productos ya que se lo puede embotellar o se lo puede almacenar en barriles, siendo estos productos diferentes en el sistema. Es por esta razón que, a pesar de que se obtenga un solo tipo de cerveza en cada orden de producción, y, tanto las botellas como los barriles contengan la misma cerveza, fue necesario permitir la obtención de uno o más productos terminados por cada orden de producción.
- La empresa se ha caracterizado por tener una relación cercana con sus clientes, haciendo sentir a cada uno de ellos parte de la cultura cervecera. Esto involucra saber que cada una de las órdenes de pedido que los clientes realizan a la cervecería sean supervisadas, desde el momento que salieron de bodega hasta que son recibidas por el cliente y el mismo se sienta satisfecho con el producto recibido. Por esta razón, el estado de entrega, es un dato relevante para las órdenes de pedido, mientras que para las órdenes de compra no lo es.
- Al ser una pequeña empresa, los directivos no manejan un proceso de contabilidad complejo. En la actualidad, los únicos datos que les son importantes conocer son los ingresos totales que reciben por las órdenes de pedidos generadas por clientes, y los egresos realizados por cada orden de compra generada a los proveedores. Estos datos son posibles obtenerlos tanto del módulo de compras como del módulo de ventas. Es por esta razón, que no fue necesario añadir un módulo de contabilidad en el modelo de datos.

- Se consideró en un principio la posibilidad de generar un reporte de ingresos vs egresos desde una fecha de inicio hasta una fecha límite. Sin embargo, esto producía reportes que dificultaban la toma de decisiones a directivos ya que no reflejaban la situación real de la empresa. De esta manera, se llegó a la conclusión que un reporte mensual era mucho más útil ya que así, con los resultados obtenidos en el mes, se podrían facilitar la tomar decisiones acerca de compras de inventario o de equipo. Además, estos reportes resultarían útiles para determinar si la empresa tiene un flujo de caja suficiente para soportar un aumento de la deuda necesaria para su propia expansión empresarial.

Por otro lado, fue posible obtener algunas conclusiones sobre la carrera, la universidad y la sociedad en general:

- Puedo concluir que la universidad, al contar con años de trayectoria y con un excelente prestigio tanto a nivel nacional como internacional, brindó todas las bases, conocimientos y habilidades clave necesarias para que uno como estudiante pueda ingresar con grandes capacidades y oportunidades al mundo globalizado y competitivo de hoy en día.
- Puedo concluir que la carrera de Ingeniería en Sistemas y Computación de la PUCE me formó como un profesional capaz de desempeñar con honradez y responsabilidad las normas legales y éticas de mi profesión como ingeniero; además que me permitió conocer las ciencias exactas necesarias para el desarrollo del pensamiento lógico, necesario para la comprensión de sistemas informáticos. Finalmente, creo que desarrolló en mí una pasión por siempre querer aprender cosas nuevas y estar siempre a la vanguardia de las nuevas tendencias no solo tecnológicas, sino también sociales y culturales.

- Después de 5 años de estudio formándome como un ingeniero de sistemas, puedo concluir que es una carrera que requiere mucho esfuerzo, mucho estudio y mucha dedicación aún después de terminada; a veces más que otras carreras, ya que el mundo tecnológico nunca deja de cambiar y no para de sorprendernos con cosas nuevas. Por esto, un ingeniero en sistemas siempre debe estar a la vanguardia y aprender sobre las nuevas tecnologías y tendencias, ya que así mantenemos buenas oportunidades de competencia y posicionamiento en este mundo de hoy en día que se ha vuelto bastante globalizado y competitivo.

6.2. Recomendaciones

- Se recomienda el uso de PostgreSQL como gestor de base de datos cuando se utiliza JAVA como lenguaje de programación, ya que al ambos ser maduros, confiables y seguros, son bastante compatibles; además que existe una herramienta madura y confiable para la conexión de ambos, como lo es el JDBC driver. Álvaro Hernández (2014) asegura en una de sus conferencias: “Probablemente, existe más código JAVA accediendo a PostgreSQL que cualquier otro lenguaje de programación”.
- Se recomienda el uso de software libre y gratuito en pequeñas y medianas empresas, ya que el objetivo principal de las mismas debería ser destinar sus recursos a la expansión de su infraestructura y/o producción, de tal manera que sus intentos de crecimiento empresarial no se vean afectados por el gasto innecesario de dinero en software con licencias. Además, es recomendado ya que las licencias de software libre permiten la instalación del software en tantas máquinas como el usuario desee, sin la necesidad de comprar una licencia para cada máquina de la empresa.

- Se recomienda que para el desarrollo de sistemas de información con tiempos de entrega relativamente cortos, se empleen metodologías ágiles ya que estas priorizan el software funcionando sobre la documentación extensiva. Siempre que sea posible, hay que reducir al mínimo indispensable el uso de documentación, que genera y demanda gran cantidad de tiempo y de trabajo que no aporta un valor directo al producto.
- Se recomienda agendar con tiempo las reuniones con el cliente para la entrega de los prototipos después de cada iteración, de forma que cuando se realice la reunión con el mismo, exista el tiempo suficiente para obtener toda la retroalimentación posible acerca del prototipo presentado. De esta manera, se podrá manejar de manera más eficiente los cambios, y el producto final que se obtenga sea uno de calidad acorde a las necesidades del usuario.
- Se recomienda controlar los datos que el usuario pueda ingresar desde la interfaz, de manera que el sistema no falle ni presente inconveniente alguno cuando el usuario ingresa un dato incorrecto o cuyo formato no es aceptado por la base de datos o el programa. Como ejemplo, no se debería permitir ingresar letras en campos que correspondan al número de teléfono, o no se debería permitir ingresar cantidades negativas cuando se compra o vende un producto.
- Se sugiere que en el análisis de requerimientos para cualquier sistema de información, se analice la obligatoriedad o no de cada uno de los atributos y campos que presenta cada entidad.
- Debido a que la empresa no poseía ninguna herramienta tecnológica que permitiera la gestión de sus principales procesos, es recomendable que cuando se

implemente el sistema en la empresa, se realice el traspaso de información con supervisión del desarrollador del sistema para asegurar que el mismo quede en un estado estable y la información sea almacenada de manera correcta.

- Se recomienda el desarrollo de interfaces amigables con el usuario, que faciliten la usabilidad del sistema, de tal manera que no sea necesaria una estricta capacitación a los usuarios para el uso del sistema, ni la elaboración de manuales de usuario. La usabilidad se refiere a la facilidad con la que el usuario puede utilizar una herramienta particular con el fin de alcanzar el objetivo para el cual el sistema o la aplicación fueron desarrollados.
- Se recomienda a los usuarios del sistema, que la información almacenada en el sistema del inventario de la empresa, sea revisada continuamente, de manera que se pueda garantizar una información actual y confiable que no presente inconvenientes al momento que el cliente realice un pedido, ni en el momento en que se desee realizar una compra a algún proveedor.
- Se recomienda a algún posible futuro desarrollador, que en caso de querer ampliar y extender el presente sistema, se mantenga y continúe con la adecuada documentación del sistema, y se continúe con la utilización de estándares de diseño que fueron implementados en el desarrollo del prototipo, de manera que la experiencia del usuario en el sistema no se vea afectada.
- Se recomienda a algún posible futuro administrador del sistema, que en el caso de un desmesurado crecimiento en la base de datos, se realicen respaldos al menos semestrales de la información; además de una actualización o renovación de credenciales de usuario para la autenticación y el ingreso al sistema.

- Se recomienda a los usuarios no perder el contacto con el administrador del sistema una vez este sea implementado, ya que a pesar de que esto puede cambiarse o arreglarse en un futuro, en un principio solo él es capaz de manejar las credenciales de usuario para la autenticación y el ingreso al sistema.
- Se recomienda a los usuarios del sistema y a los directivos de la empresa guardar la documentación correspondiente del sistema desarrollado, ya que si en un futuro desean extender o ampliar el sistema, esta resultará muy útil y facilitará este trabajo a cualquiera que sea el desarrollador.
- Se recomienda a cualquier ingeniero en sistemas o carreras afines relacionadas con la tecnología, que nunca se pierdan las ganas de aprender. La tecnología es un área que siempre está en constante renovación, crecimiento e innovación, por lo que siempre es y será necesario aprender nuevas cosas para poder estar a la altura de lo que el mundo requiere.
- Como recomendación final, puedo sugerir que en trabajos de desarrollo se formen de preferencia equipos interdisciplinarios, en los que incluso se tome en cuenta al cliente como parte del equipo de trabajo. De esta manera, los procesos desarrollados llegan a ser muy creativos, y amplían las posibilidades de éxito de cualquier proyecto. El trabajo en un equipo interdisciplinario incluso implica más capacidad para dialogar e intercambiar opiniones, lo cual resulta una característica fundamental cuando empleamos sobretodo metodologías ágiles.

Referencias

- Bonilla Huerta, E., & Ramírez Cruz, F. (2014). *Advances in Intelligent Information Technologies*. México: Instituto Politécnico Nacional.
- Camuña, J. F. (2014). *Lenguajes de definición y modificación de datos SQL (UF1472)*. Madrid: IC Editorial.
- El Proceso contable de una empresa*. (25 de Diciembre de 2016). Obtenido de Finanzas y Contabilidad: <https://finanzascontabilidad.com/proceso-contable-empresa/>
- Flórez Fernández, H. (2012). *Programación orientada a objetos usando java*. Bogotá: Ecoe Ediciones.
- Hernández, Á. [Postgres Open SV 2018]. (24 de Noviembre de 2014). PostgreSQL and Java: Best Practices - Alvaro Hernandez. [Archivo de vídeo]. Obtenido de https://www.youtube.com/watch?v=i93hFl3n_e8
- Kniberg, H., & Skarin, M. (2014). *KanBan y Scrum – obteniendo lo mejor de ambos*. C4Media Inc.
- López, J. L. (2014). *Programación orientada a objetos C++ y Java: un acercamiento interdisciplinario*. Grupo Editorial Patria.
- Maldonado, J. A. (2011). *Gestión de procesos*. Madrid: B - EUMED.
- Morán Jorquera, W. M. (2018). *Desarrollo de un Sistema de Información para la Gestión de Producción en Empresa de Cervecería Artesanal aplicando KanBan*. Quito: PUCE.
- Net-informations.com. (2017). *What is Java virtual machine?* Obtenido de Net-informations: <http://net-informations.com/java/intro/jvm.htm>
- Palmer, J. (2006). *How to brew : everything you need to know to brew beer right the first time*. Boulder: Brewers Publications.
- Real Academia Española. (2001). *Diccionario de la lengua española (22.a ed.)*. Obtenido de <http://dle.rae.es>
- Ruiz Rodríguez, R. (2011). *Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java*. Córdoba: El Cid Editor.

Sociedad Ecuatoriana de cerveceros artesanales SECA. (s.f.). *Curso Básico de elaboración de Cerveza Artesanal*. Obtenido de Sociedad Ecuatoriana de Cerveceros Artesanales: <http://secaecuador.es.tl/Home.htm>

Valderrey Sanz, P. (2014). *Administración de sistemas gestores de bases de datos*. Madrid: RA-MA Editorial.