

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**

**“CONTROL DE LA ELABORACIÓN DE PRODUCTOS LÁCTEOS”**

**CASO: FLOR DE LECHE S.R.L.**

PARA ÓPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: JUNIOR MANUEL MUÑOZ CUIZA**

**TUTORA METODOLÓGICA: LIC. MENFY MORALES RÍOS**

**ASESOR: LIC. MARCELO GERMÁN ARQUIPA CHAMBI**

LA PAZ – BOLIVIA

2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

### *DEDICATORIA*

*El presente trabajo es dedicado a mi madre con mucho cariño, quien está apoyándome constantemente, a mi padre que desde el cielo está siempre alentándome seguir adelante y a mis queridos hermanos que siempre puedo contar con su apoyo incondicional.*

## **AGRADECIMIENTO**

Ante todo a Dios por su bendición

Para poder realizar este proyecto de la mejor manera posible, fue necesario el apoyo de muchas personas a las cuales quiero agradecer.

A mí querida tutora Lic. Menfy Morales, por su valioso asesoramiento y apoyo para la culminación de mi proyecto de grado. Gracias por su colaboración.

A mí querido asesor Lic. Marcelo Aruquipa, a quien le debo el hecho de que este Proyecto de Grado se haya ejecutado satisfactoriamente. Gracias por sus consejos y orientación.

A mis padres †Victor Muñoz Lamas(QEPD) y Magali Cuiza Morales, quienes han sido un apoyo moral y económico para lograr este fin. Gracias por su paciencia.

A mis hermanos Claudia e Israel, por el apoyo incondicional. Gracias por su paciencia.

A mi enamorada Jazmin, quien me apoyo y alentó incondicionalmente y a mis amigos, a cada uno de ellos que me demostraron su apoyo en cada situación difícil.

## **RESUMEN**

El control de la producción no constituye un fin en sí misma, sino un medio para alcanzar la eficacia y eficiencia de las empresas, a través del trabajo de las personas, y para establecer condiciones favorables que les permitan conseguir los objetivos deseados.

La empresa Flor de Leche SRL., fue constituida legalmente en 1998 como una empresa unipersonal, uno de los objetivos que pretende alcanzar es “Mejorar la calidad de los quesos: quesos defectuosos no superan los 200 kilos por año desde 01-01-2016 (300 kilos desde 01-01-2015).”

El presente proyecto pretende coadyuvar a este fin, a través de la implementación de un sistema, el cual permite realizar el control eficiente de la producción y pH de los productos lácteos procesados, así ofreciendo información ordenada, completa, oportuna para la toma de decisiones.

De esta manera el Sistema de Información para el control de productos lácteos procesados aporta un control transparente, eficiente, eficaz y moderno. Para este propósito se utiliza los recursos tecnológicos y métodos apropiados para su desarrollo e implementación del proyecto presentado.

## **ABSTRACT**

The production control is not an ending in itself, but it means to achieve effectiveness and efficiency of enterprises, through the work of people, and to establish favorable conditions to enable them to achieve the desired objectives.

The Flor de Leche SRL company was legally constituted in 1998 as a sole proprietorship, one of the objectives pursued is "Improving the quality of the cheese: cheeses defective exceeding 200 kilos per year from 01-01-2016 (300 kilos from 01-01-2015). "

This project aims to contribute to this ending, through the implementation of a system, which allows efficient control of the production and pH of processed dairy products, offering tidy, complete, timely information for decision-making information.

In this way the information system for the control of processed dairy products provides a transparent, efficient, effective and modern control. For this purpose the technological resources and appropriate methods for development and implementation of the project submitted is used.

## INDICE

CAPITULO I.....	12
MARCO REFERENCIAL .....	12
1.1. INTRODUCCIÓN.....	12
1.2. ANTECEDENTES .....	12
1.2.1. Trabajos UMSA.....	12
1.2.2. Trabajos Internacionales.....	13
1.2.3. Antecedentes de la Institución Flor de Leche SRL. ....	13
1.3. PROBLEMÁTICA .....	14
1.3.1. PLANTEAMIENTO DEL PROBLEMA .....	15
1.3.2. Problemas Secundarios.....	15
1.4. OBJETIVOS.....	15
1.4.1. Objetivo General.....	15
1.4.2. OBJETIVOS ESPECÍFICOS .....	16
1.5. JUSTIFICACIÓN.....	16
1.5.1. Técnica.....	16
1.5.2. Social .....	16
1.5.3. Económica .....	16
1.6. LÍMITES Y ALCANCES .....	17
1.6.1. Límites .....	17
1.6.2. Alcances.....	17
1.7. METODOLOGÍA .....	17
1.7.1. Personal SCRUM .....	17
1.8. APORTES .....	18
CAPITULO II.....	19
MARCO TEORICO .....	19
2.1. INTRODUCCIÓN .....	19
2.2. LA INSTITUCIÓN.....	19
2.2.1. Misión.....	20

2.2.2.	Visión .....	20
2.2.3.	Objetivos.....	20
2.2.4.	Estructura organizacional de Flor de Leche S.R.L.....	21
2.3.	POTENCIAL DE HIDROGENO (PH).....	21
2.4.	RENDIMIENTO DE PRODUCCIÓN DE QUESO Y YOGURT.....	22
2.5.	METODOLOGÍA DE DISEÑO UML.....	23
2.5.1.	Diagrama de casos de uso.....	24
2.5.2.	Diagrama de clases .....	24
2.5.3.	Diagrama de secuencia .....	25
2.5.4.	Diagrama de estados.....	26
2.5.5.	Diagrama de paquetes.....	26
2.6.	METODOLOGÍA DE DESARROLLO ÁGIL SCRUM.....	27
2.6.1.	Fases de SCRUM .....	27
2.6.1.1.	Planes de lanzamiento .....	27
2.6.1.2.	Distribución, revisión y ajuste de los estándares del producto.....	28
2.6.1.3.	Sprint .....	28
2.6.1.4.	Revisión Sprint .....	29
2.6.1.5.	Cierre .....	30
2.7.	ARQUITECTURA DE SOFTWARE EN EL CICLO DE DESARROLLO DE SCRUM .....	30
2.8.	EVALUACIÓN DE ARQUITECTURAS DE SOFTWARE CON ATAM.....	31
2.8.1.	Fase 1:.....	32
2.8.2.	Fase 2:.....	33
2.9.	REQUERIMIENTO DE SOFTWARE Y HARDWARE.....	34
2.9.1.	Java.....	34
2.9.1.1.	Librerías Java.....	34
2.9.2.	PostgreSQL.....	35
2.10.	PRUEBA .....	35
2.10.1.	Prueba Unitaria.....	35
2.11.	CALIDAD DEL SOFTWARE .....	36
2.11.1.	Funcionalidad .....	37
2.11.2.	Confiabilidad .....	38

2.11.3. Mantenimiento.....	39
2.11.4. Certificación del software.....	39
2.11.5. Medición del software .....	39
2.12. ESTIMACIÓN DE COSTOS .....	40
2.12.1. COCOMO.....	40
- MODELOS DE ESTIMACIÓN.....	40
2.13. SEGURIDAD DEL SISTEMA .....	42
2.13.1. Controles de acceso .....	43
2.13.2. Identificación y Autenticación .....	43
2.13.3. Roles.....	43
CAPITULO III .....	44
MARCO APLICATIVO .....	44
3.1. INTRODUCCIÓN.....	44
3.2. PLANES DE LANZAMIENTO.....	44
3.3. SPRINT 0 .....	44
3.3.1. Análisis de requisitos preliminares.....	45
3.3.2. Diseño de la arquitectura .....	47
3.3.2.1. Diagrama de casos de uso.....	47
3.3.2.2. Diagrama de clases .....	49
3.3.2.3. Diagrama de secuencia .....	52
3.3.2.4. Diagrama de estados.....	54
3.3.2.5. Diagrama de paquetes.....	55
3.3.3. Evaluación de la arquitectura .....	55
3.4. SISTEMA ESQUELETO O PROTOTIPO.....	56
3.5. PLANES DE LANZAMIENTO DE ACUERDO A LOS NUEVOS SPRINTS .....	57
3.6. SPRINT 1 .....	57
3.6.1. Sprint Backlog 1 .....	58
3.7. SPRINT 2 .....	58
3.7.1. Prueba de Unidad .....	59
3.7.2. Sprint Backlog 2.....	60



3.8.	SPRINT 3 .....	61
3.8.1.	Prueba de Unidad .....	63
3.8.2.	Sprint Backlog 3 .....	64
3.9.	SPRINT 4 .....	64
3.9.1.	Prueba de Unidad .....	64
3.9.2.	Sprint Backlog 4 .....	65
3.10.	SPRINT 5 .....	65
3.10.1.	Sprint Backlog 5 .....	66
3.11.	SPRINT 6 .....	66
3.11.1.	Sprint Backlog 6 .....	68
3.11.2.	Diagrama navegacional .....	68
CAPITULO IV .....		70
PRUEBAS Y EVALUACIÓN DEL SISTEMA .....		70
4.1.	INTRODUCCIÓN .....	70
4.2.	CALIDAD DE SOFTWARE .....	70
4.2.1.	Confiability .....	72
4.2.2.	Mantenimiento.....	73
4.3.	ESTIMACIÓN DE COSTOS CON COCOMO .....	73
4.4.	SEGURIDAD DEL SISTEMA .....	75
4.4.1.	Controles de acceso .....	75
4.4.2.	Identificación y autenticación.....	75
4.4.3.	Roles .....	76
CAPITULO V .....		77
CONCLUSIONES Y RECOMENDACIONES .....		77
5.1.	INTRODUCCIÓN .....	77
5.2.	CONCLUSIONES.....	77
5.3.	RECOMENDACIONES .....	78

## INDICE DE TABLAS

Tabla 2.1. Rangos del nivel de pH .....	21
Tabla 2.2. Tipos de reuniones de Scrum .....	29
Tabla 2.3. Elementos de Scrum.....	29
Tabla 2.4. Factor de ponderación .....	37
Tabla 2.5. Constantes para calcular distintos aspectos de costes. ....	41
Tabla 2.6. Constantes a reemplazar en la fórmula.....	42
Tabla 3.1. Planes de lanzamiento del Sprint 0.....	44
Tabla 3.2. Lista de requerimientos funcionales del sistema. ....	46
Tabla 3.3. Requerimientos de Hardware .....	46
Tabla 3.4. Requerimientos de Software.....	46
Tabla 3.5. Registrar datos de producción .....	48
Tabla 3.6. Validación de reportes. ....	48
Tabla 3.7. Establecer personal de producción. ....	49
Tabla 3.8. Planes de lanzamiento para los diferentes Sprints.....	57
Tabla 3.9. Primer sprint backlog realizado.....	58
Tabla 3.10. Segundo sprint backlog realizado.....	61
Tabla 3.11. Tercer sprint backlog realizado. ....	64
Tabla 3.12. Cuarto sprint backlog realizado.....	65
Tabla 3.14. Sexto sprint backlog realizado.....	68
Tabla 4.1. Cálculo de cuenta total de la métrica de punto función.....	70
Tabla 4.2. Valor de ajuste de complejidad de punto de función. ....	71
Tabla 4.3. Escala de Punto de Función.....	72
Tabla 4.4. Detalle de costos.....	75

## INDICE DE FIGURAS

Figura 1.1. Organigrama de Gerencia de Producción (GPR).....	14
Figura 2.1. Formula del rendimiento del queso y yogurt. ....	22
Figura 2.2. Representación gráfica del rendimiento. ....	23
Figura 2.3. Modelado de casos de uso.....	24
Figura 2.4. Diagrama de clases.....	25
Figura 2.5. Diagrama de secuencia.....	25
Figura 2.6. Diagrama de estado.....	26
Figura 2.7. Diagrama de paquetes .....	27
Figura 2.8. Propuesta de gestión de la arquitectura de software en Scrum. ....	31
Figura 3.1. Modelado de casos de uso.....	47
Figura 3.2. Diagrama de clases.....	50
Figura 3.3. Diagrama Entidad-Relación. ....	51
Figura 3.4. Diagrama Relacional.....	52
Figura 3.5. Diagrama de registro de datos de producción. ....	53
Figura 3.6. Diagrama de validación de reportes.....	53
Figura 3.7. Diagrama del registro de productos. ....	54
Figura 3.8. Diagrama de estados. ....	54
Figura 3.9. Diagrama de paquetes. ....	55
Figura 3.10. Ingreso al sistema.....	56
Figura 3.11. Sistema base.....	57
Figura 3.12. Ventana de despliegue de tablas elaboradas. ....	58
Figura 3.13. Ventana de transacciones de grupos. ....	59
Figura 3.14. Ventana de manejo de permisos o privilegios. ....	59

Figura 3.15. Pruebas del sprint 2 .....	60
Figura 3.16. Ventana de transacciones de tipos de producto.....	61
Figura 3.17. Ventana de transacciones de productos.....	62
Figura 3.18. Ventana de transacciones de fabricación .....	62
Figura 3.19. Pruebas del sprint 3 .....	63
Figura 3.20. Opción de exportación a Excel. ....	64
Figura 3.21. Pruebas del sprint 4 .....	65
Figura 3.22. Reporte generado por fecha de selección.....	66
Figura 3.23. Gráficas de rendimientos.....	67
Figura 3.24. Reporte en formato Word.....	67
Figura 3.25. Gráfica de pH. ....	68
Figura 3.26. Diagrama navegacional.....	69
Figura 4.1. Inserción de datos para COCOMO .....	74
Figura 4.2. Resultado de estimación de costos .....	74

# CAPITULO I

## MARCO REFERENCIAL

### 1.1. Introducción

La necesidad de contar con tecnologías de información que en un principio permitieron al ser humano comunicarse y facilitar la ejecución de algunas actividades como se va reflejando hoy en día en todas las instituciones, empresas, fabricas, y otros, tienen la necesidad de manejar su información de forma segura, confiable, y que siempre este a su disposición en el momento que ellos más lo requieran, así de esta manera ellos pueden consultar la información, generar reportes, validar información, y de alguna manera tomar decisiones muy importantes.

Es así como la empresa Flor de Leche SRL., se dedica a la producción de lácteos de alta calidad, quienes consideran la utilidad de la información muy importante a la hora de la toma de decisiones.

En el proceso de fabricación, el control de registro de datos es de forma manual, por lo tanto, se plantea realizar el proceso de automatización, que está de acuerdo a normas de la empresa, pero para finalizar los procesos, la Gerencia desea conocer que tanto se obtuvo de Queso y Yogurt, a lo que llamamos cuanto es el rendimiento, y analizar el potencial de Hidrogeno (pH). De esta manera conseguir información oportuna para obtener los rendimientos y el pH, y tomar decisiones respecto a la calidad del proceso.

En función a lo mencionado, se plantea una vía alternativa que permita efectivizar el proceso de obtener información sobre la producción de lácteos en cada proceso.

### 1.2. Antecedentes

#### 1.2.1. Trabajos UMSA

En la carrera de informática de la Universidad Mayor de San Andrés (UMSA) se encontraron proyectos de grado relacionados al control de producción como ser:

- Sistema de Información para la Producción Planificada DELIZIA [ANM06], tiene el objetivo de implantar un Sistema de Información que permita mejorar la planificación de la producción en sus puntos más críticos: ajuste de presupuesto, gestión de almacén con el método ponderado y el seguimiento de producción, aplicando la metodología del El Lenguaje de Modelado Unificado (UML), donde

se captura los requisitos de los sistemas a través de paquetes transaccionales llamados casos de uso.

- Sistema de Almacenes y Producción SOCOVIAL [MBA97], tiene el objetivo de diseñar e implementar un sistema de información basado en el control de inventarios y mejorar el control productivo, aplicando las metodologías de Análisis Estructurado, la aplicación de un método de inventarios acorde a las necesidades y el diseño orientado a objetos de Martín & Odell, el sistema contempla procesos manuales en un 70% y el 30% restante es información automatizada, que se usa para emitir todas las ordenes correspondientes.
- Sistema de Información Integrado PIL-CHUQUISACA S.A.(SII-PIL Chuquisaca) [ACM01], tiene como objetivo controlar y optimizar el proceso de la información en la planta Industrializadora. Además de realizar un análisis de diseño del nuevo sistema, con un modelo de implantación del usuario y del sistema, aplicando la metodología de Análisis y Diseño Estructurado, la cual parte el sistema funcionalmente y según distintos comportamientos se establece la esencia de lo que se debe construir, donde las herramientas de modelado son el Modelo Esencial que a la vez se divide en el Modelo Ambiental y el Modelo de Comportamiento.

### **1.2.2. Trabajos Internacionales**

- Tesis, Modelo matemático para predecir el rendimiento del queso a partir de la composición química de la dieta[MMPR12], tiene como objetivo elaborar un modelo matemático que permita predecir el rendimiento de queso fresco a partir de las características nutrimentales de la dieta en ganado bovino usando para ello un análisis sistemático de datos derivados de artículos del Journal of Dairy Science.

### **1.2.3. Antecedentes de la Institución Flor de Leche SRL.**

Flor de Leche nació como empresa unipersonal en 1998 en el municipio de Achocalla que se encuentra entre La Paz y El Alto. Actualmente Flor de Leche compra la leche de más de 150 familias y emplea a más de 36 personas.

El siguiente organigrama (ver figura 1.1.), el cual es un fragmento del organigrama general (ver Anexo A), representa en que área se desenvuelve el sistema de producción.

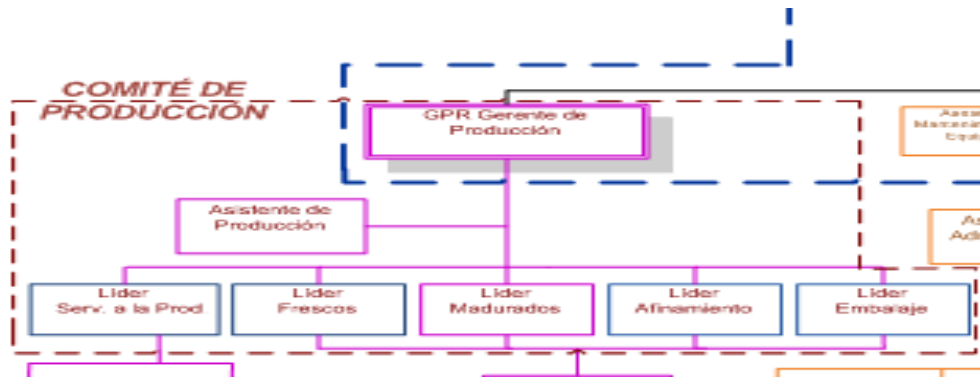


Figura 1.1. Organigrama de Gerencia de Producción (GPR)

Fuente: [FDL16]

### 1.3. Problemática

La empresa Flor de Leche S.R.L., se dedica actualmente a la industrialización de productos lácteos, donde en su primera etapa es la recepción de leche, en este proceso se cuenta con tres trabajadores quienes verifican la calidad de la leche y la acidez, son muy importantes estos parámetros ya que podrían afectar a los rendimientos de producción de Quesos y Yogurt, y el potencial de Hidrogeno (pH), luego sigue con la cantidad de kilos de leche a pasteurizar y el proceso de traslado de leche pasteurizada a la fabricación, continuando con el proceso de la fabricación del queso o yogurt donde a menudo se cuenta de tres a cinco trabajadores quienes producen el producto lácteo, al finalizar este proceso se realiza la anotación de la masa del producto final, además de contar los moldes o frascos, el siguiente proceso viene a ser el control de calidad donde dos personas expertas en calidad son responsables del área, donde se calcula el nivel de pH y finalizando el cálculo de rendimientos.

Flor de Leche S.R.L. actualmente cuenta con varias planillas Excel que utiliza Gerencia de Producción (GPR), estas planillas son utilizadas para calcular los rendimientos, informar cual personal fabrico el lácteo, graficar los rendimientos en sentido inverso, conocer las gráficas de pH, diariamente, semanalmente, mensualmente, y anualmente además de realizar comparaciones con años pasados. Utilizar planillas Excel es muy adaptable, pero tiene sus limitaciones, existen varios informes que Excel normalmente no lo puede realizar, además de tener varias planillas Excel, puede producir una fácil pérdida de información. Usualmente agregar datos a las planillas es muy moroso, ya que se manejan aproximadamente 90 procesos mensualmente, lo cual insertar cada vez resulta ser molesto, además los datos se repiten más de dos veces en varias planillas, produciendo redundancia en la información, entonces los

datos no son íntegros. La mayoría de los informes están basados en tablas dinámicas de Excel, y de estas mismas generando las gráficas, las cuales al momento de hacer comparaciones anuales, se demora demasiado en armar los informes, generando molestias por parte de Gerencia General (GGE), esta información es vital, ya que permite controlar con más seguridad el proceso de fabricación.

### **1.3.1. Planteamiento del problema**

Por lo expuesto anteriormente se plantea el siguiente problema:

¿Cómo mejorar el manejo de la información acerca del rendimiento de producción y pH de productos lácteos en la institución Flor de Leche SRL. que posibilite la reducción de tiempo y obtención de información confiable y oportuna?

### **1.3.2. Problemas Secundarios**

- El proceso de conseguir información acerca del rendimiento es bastante moroso para gerencia.
- El sistema de producción actual está diseñado en plantillas Excel, haciéndolo muy inseguro y con bastantes limitaciones.
- Los informes están diseñados en tablas dinámicas de Excel, y como Excel solo permite algunas opciones de gráficas, se tiene que limitar la representación gráfica de la información para gerencia.
- Actualmente existen varias plantillas Excel que se utiliza en gerencia de producción, donde fácilmente se pueden perder la información.
- El proceso de agregar datos en el sistema actual, es muy moroso, donde usualmente los datos se repiten, generando que se vuelvan a llenar los datos más de dos veces, y existiendo redundancia de datos.
- Los datos no son íntegros, debido a que se encuentran en varias planillas de Excel, muchas de las planillas contienen información redundante.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

- Desarrollar un sistema para realizar el control eficiente de la producción y pH de los productos lácteos procesados.



#### **1.4.2. Objetivos Específicos**

- Implementar tecnologías de arquitectura de software para desarrollar un sistema escalable.
- Obtener informes confiables, donde se puedan hacer comparaciones con datos pasados, de esta manera pronosticar los rendimientos, y tomar buenas decisiones.
- Unir toda la información de todas las planillas para lograr integrar los datos y de esta manera no perder información.
- Agilizar el llenado de datos, para lograr mejores tiempos en el trabajo de gerencia de producción.
- Integrar la información en un solo motor de base de datos confiable.
- Diseño e implementación de una Base de Datos.

#### **1.5. Justificación**

##### **1.5.1. Técnica**

El proyecto es justificable técnicamente debido a las facilidades en cuanto a tecnología cuenta Flor de Leche S.R.L., tanto en hardware como en software son reutilizados para la implementación del presente proyecto.

Es importante mencionar que el desarrollo del proyecto es realizado en su totalidad, con herramientas de software libre, lo cual permite un costo mínimo en el proyecto.

##### **1.5.2. Social**

El presente proyecto se convierte en un instrumento que facilite a la toma de decisiones, permitiendo a la empresa Flor de Leche S.R.L. contar con un mejor proceso a la hora de tomar decisiones.

Adicionalmente, los clientes o consumidores de los productos lácteos son beneficiados al poder contar con productos de buena calidad.

##### **1.5.3. Económica**

El presente proyecto es económicamente justificable debido a que los reportes son generados ágilmente, y además de agilizar el trabajo de Gerencia de Producción, quienes pueden poner más énfasis en el área de producción y de esta manera se beneficia Flor de Leche SRL.

## **1.6. Límites y Alcances**

### **1.6.1. Límites**

El presente proyecto, no se desenvuelve en las áreas de contabilidad, administración, recurso humanos, permisos, inventarios los cuales no son contemplados en este primer módulo.

### **1.6.2. Alcances**

El presente proyecto, aparte de ser escalable, para una primera versión, solo permite obtener datos acerca de los rendimientos, pH, desviaciones estándar por producto, promedios de rendimientos y pH, además de representar graficas acerca de los mismos datos, permitiendo a Gerencia de producción tomar decisiones.

## **1.7. Metodología**

Los métodos preliminares utilizados para determinar los requerimientos de los usuarios y los procesos a automatizar son:

- Entrevistas personales con el encargado de la gerencia de producción.
- Análisis de la situación actual.
- Método ágil SCRUM.

### **1.7.1. Personal SCRUM**

La metodología a utilizar en el presente proyecto, es el Personal Scrum, o Scrum para una persona.

La metodología Personal Scrum es una excelente herramienta para realizar trabajos.

Personal Scrum es una metodología ágil que está adaptada y aplicada para prácticas Scrum en un proyecto y una persona. Esto promueve una productividad personal a través de la observación, adaptación, elaboración progresiva, priorizando el tamaño del proyecto y el tiempo de trabajo.

En la metodología Scrum, solo una persona podría tener tres roles, esto no es difícil, porque solo uno tiene la responsabilidad todo el tiempo.

La alianza Scrum provee cuatro actividades:

- Primero: Planeando un Sprint.
- Segundo: Scrums diariamente.
- Tercero: Revisiones Sprint.
- Cuarto: Retro alimentación Sprint.

Aparte de las actividades se tiene tres artefactos usados en Scrum:

- Primero: Producto Backlog.
- Segundo: El Sprint backlog.
- Tercero: El Burndown chart.

Cada uno de estos artefactos debe ser usado de la misma manera como el Manager en Personal Scrum. [PJ11]

### **1.8. Aportes**

Los aportes que ofrece el proyecto a la empresa es automatizar sus procesos rutinarios, minimizar y optimizar tiempos de ejecución generando información que coadyuve a la fácil y correcta toma de decisiones de la empresa Flor de Leche SRL.

El aporte académico en la implementación del proyecto es utilizar una metodología para evaluar la arquitectura de software, llamada Architecture Tradeoff Analysis Method (ATAM) la cual permite evaluar la arquitectura de software en una fase inicial en la metodología de desarrollo SCRUM. La metodología es muy necesaria aplicarla, ya que no se debería hacer un diseño en pleno desarrollo, entonces ATAM especifica tener una arquitectura correctamente diseñada antes de iniciar los ciclos de desarrollo.

Además aportar académicamente con la librería que está desarrollado en Java, la cual permite generar consultas, altas, bajas, modificaciones, conexiones, ejecución de scripts en el servidor de Postgres, los estudiantes pueden hacer uso de ella para desarrollar más rápidamente sistemas de información que vinculen entre JAVA y Postgres.

Así también se aporta con la librería de exportación de datos a Excel, la librería permite generar cabeceras, datos y manejar el tipo de datos que se requiera, para ello el estudiante puede hacer uso de ella en cualquier momento, y de alguna manera acelerar el desarrollo de sistemas.

## **CAPITULO II**

### **MARCO TEORICO**

#### **2.1. Introducción**

En este capítulo se hace una descripción del marco conceptual y del marco teórico necesario para implementar el proyecto.

El marco conceptual es una serie de ideas o conceptos coherentes organizados de tal manera que sean fáciles de comunicar a los demás sobre el desarrollo del proyecto.

El marco teórico es la etapa en que reunimos información documental para confeccionar el diseño metodológico de la investigación, describiendo la conceptualización de la metodología SCRUM para el desarrollo del proyecto, además de detallar cada flujo de trabajo del ciclo de vida como: requerimiento de software y hardware, implementación, pruebas, mantenimiento del sistema y calidad del software.

#### **2.2. La institución**

Es una empresa boliviana que produce excelentes quesos, cremas de queso y yogurts de manera artesanal y natural, con un control riguroso de calidad y de higiene. La leche proviene del Altiplano, las tierras de pasto más altas del mundo, que les proporcionan un carácter único. Las recetas de los quesos son originarias de Europa.

Flor de Leche nació como empresa unipersonal en 1998 en el municipio de Achocalla que se encuentra entre La Paz y El Alto. Actualmente Flor de Leche compra la leche de más de 150 familias y emplea a más de 36 personas.

Flor de Leche trabaja según los principios de responsabilidad social y de conciencia ambiental. Busca ser una empresa con responsabilidad social, conciencia medio ambiental y rentabilidad económica. Aspira a promover el desarrollo local en estos ámbitos a través del emprendimiento de una empresa sostenible, que permita irradiar modelos y propuestas para otras iniciativas. Considera el factor humano como primordial, desarrollando relaciones de respeto y valoración mutua entre las diversidades humanas y fomentando su participación y su capacitación.

Flor de Leche cree en el potencial de la producción artesanal de alta calidad con uso intensivo de mano de obra como contribución al desarrollo regional en base a las condiciones del país.

Es una empresa privada que participa activamente en el desarrollo sostenible con visión social y ecológica.

La leche que se compra cada día, es analizada, clasificada y almacenada bajo estricto control de calidad, higiene y temperatura. Esta leche es usada el mismo día en la producción de 42 tipos de productos que se pueden clasificar en varios grupos: quesos madurados prensados, quesos madurados florecidos, cremas de queso, quesos frescos, yogurts y ocasionalmente otros productos.

Flor de Leche con sede central en Achocalla, provee de sus productos a los departamentos de Santa Cruz, Cochabamba y La Paz, además de proveer productos al subsidio,

Los productos se venden en tiendas especializadas y en supermercados. El empaquetado de los mismos, se realiza manualmente y se utilizan etiquetas y envases que incluyen precintos de seguridad y fecha de vencimiento.[FDL13]

### **2.2.1. Misión**

Ser una emprendimiento eco social, integralmente responsable y sostenible, que busca interactuar e inspirar a nivel local y nacional para una mejor calidad de vida. [FDL16]

### **2.2.2. Visión**

Ser un emprendimiento y ejemplo eco social en permanente búsqueda de mejorar vidas, ofrecemos productos lácteos artesanales con un carácter único. [FDL16]

### **2.2.3. Objetivos**

- Cambiar el equilibrio entre subsidio de gobierno y ventas tradicionales 30%/70% al 1-1-2016 (1-1-2015 55%/45%).
- Ser un participante activo en la red de emprendedores ecos sociales la cual logra incidir en políticas públicas a 1-1-2016.
- Mejorar la calidad de los quesos: quesos defectuosos no superan los 200 kilos por año desde 1-1-2016 (300 kilos desde 1-1-2015).
- Mantener la producción en un máximo 3.000 litros de leche en la planta de Achocalla, aumentando la proporción de productos sanos y socialmente asequibles a un 5% hasta 1-1-2017.
- Fidelizar el personal a través de un programa de comunicación que es consistente, sostenido y creciente a nivel individual y colectivo.

[FDL16]



#### 2.2.4. Estructura organizacional de Flor de Leche S.R.L.

La estructura de la empresa Flor de Leche S.R.L. está estructurada por:

El Gerente Propietario quien es un participante activo en la red de emprendedor eco social la cual logra incidir en políticas públicas. El Gerente Operativo quien fortalece la gestión administrativa (transparentar contabilidad). Área de Producción la cual mejora la calidad de los quesos. Área de Comercialización quienes están encargados de hacer un análisis de mercado y de competidores, hacer un plan de marketing, mejorar, comunicar la imagen y conocimiento de Flor de Leche. Área de Recursos Humanos quienes deben fidelizar el personal a través de un programa de comunicación que es consistente, sostenido y creciente a nivel individual y colectivo. Área de Administración quienes asumen la responsabilidad directa de la contabilidad fiscal e interna.

El organigrama de Flor de Leche S.R.L. es representado gráficamente en el Anexo A.

#### 2.3. Potencial de Hidrogeno (pH)

El potencial Hidrógeno (pH) es una forma convencional y muy conveniente de expresar según una escala numérica a dimensional, el grado de acidez o basicidad de soluciones acuosas diluidas. En realidad una medida de la actividad de los iones hidrógeno en una solución electrolítica.[PHCA08]

La escala de pH es de 0 a 14. La disolución neutral, tiene un pH de 7, valores menores de 7 indican una disolución acida y valores superiores a 7 indican una disolución alcalina.

La tabla 2.1. muestra los datos en rangos de algunos productos con su pH:

Rango de pH	Alimento	pH
Acidez baja (pH 7 a 5,5)	Leche	6,3 a 6,5
	Pollo	5,6 a 6,4
	Pescado	6,6 a 6,8
	Patatas	5,6 a 6,2
Acidez media (pH 5,5 a 4,5)	Plátanos	4,5 a 5,2
	Vegetales fermentados	3,9 a 5,1
Ácidos (pH 4,5 a 3,7)	Mayonesa	3 a 4,1
	Tomates	4
Alta acidez (pH < 3,7)	Cítricos	3 a 3,5
	Manzanas	2,9 a 3,3

Tabla 2.1. Rangos del nivel de pH

Fuente: [IMPH00]

## 2.4. Rendimiento de producción de Queso y Yogurt

El rendimiento es la cantidad máxima de queso que se pueden fabricar con un volumen determinado de leche. La leche es obviamente la materia prima principal para la elaboración de los quesos. Siempre partiremos de leche natural, desnatada total o parcialmente, de la nata del suero de mantequilla o de una mezcla de algunos o de todos estos productos. [TERE10]

La leche generalmente procede de vacas, ovejas, cabras, obteniéndose quesos puros. Dependiendo del origen, así será el resultado final del queso, pudiendo variar tanto su sabor como su textura. Los quesos más suaves son los que están elaborados con leche de vaca y los más fuertes o madurados son sobre todo los quesos de oveja.

Si se utiliza la leche cruda, es decir, sin tratar, el queso conserva más su sabor y toda su grasa. La leche pasteurizada es aquella que se somete a un elevado efecto de temperatura, destruyéndose así las bacterias y gérmenes dañinos, sin alterar su composición y cualidades.

Leche de buena calidad asegura la obtención de quesos de buena calidad. Existen factores físico-químicos y microbiológicos, que afectan la coagulación de la leche y que están ligados a su composición (cantidad de proteínas solubles, balance salino, pH, y otros.) por otro lado la carga microbiana por razones obvias afecta la calidad sanitaria, la inocuidad del queso y la vida útil del mismo.

El proceso inicia con el tratamiento de la leche (pasteurización), siguiendo con la producción de queso o yogurt, y finalmente el pesado del queso o el volumen de Yogurt. [ELQU04]

El rendimiento se calcula mediante la fórmula, ver figura 2.1.

$$\text{Rendimiento} = \frac{\text{kilos de leche a producir}}{\text{kilos de Queso producido}} \quad \text{Rendimiento} = \frac{\text{kilos de leche a producir}}{\text{kilos de Yogurt producido}}$$

Figura 2.1. Formula del rendimiento del queso y yogurt.

Fuente: [FFDLQY11]

Se representa el rendimiento en una forma gráfica con la figura 2.2.



Figura 2.2. Representación gráfica del rendimiento.

## 2.5. Metodología de diseño UML

El Lenguaje Unificado de Modelado pre-escribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.[MSO15]

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura re-visión.



Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones y otros)
- Relaciones: Relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.[MSO15]

### 2.5.1. Diagrama de casos de uso

El diagrama de casos de uso representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la figura 2.3., se muestra un ejemplo de casos de uso, donde se muestran los actores y las operaciones que pueden realizar.[MSO15]

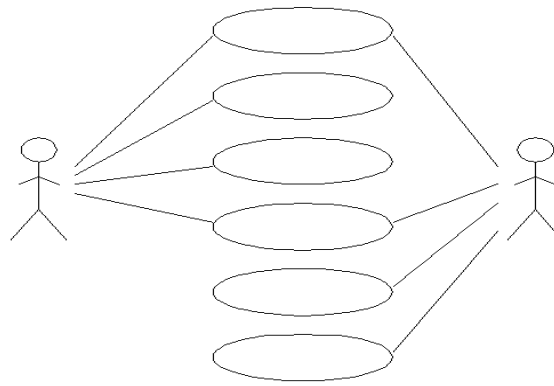


Figura 2.3. Modelado de casos de uso

Fuente: [MSO15]

### 2.5.2. Diagrama de clases

El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Este es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. En la figura 2.4., se muestran las clases globales, sus atributos y los métodos que la acompañan.[MSO15]

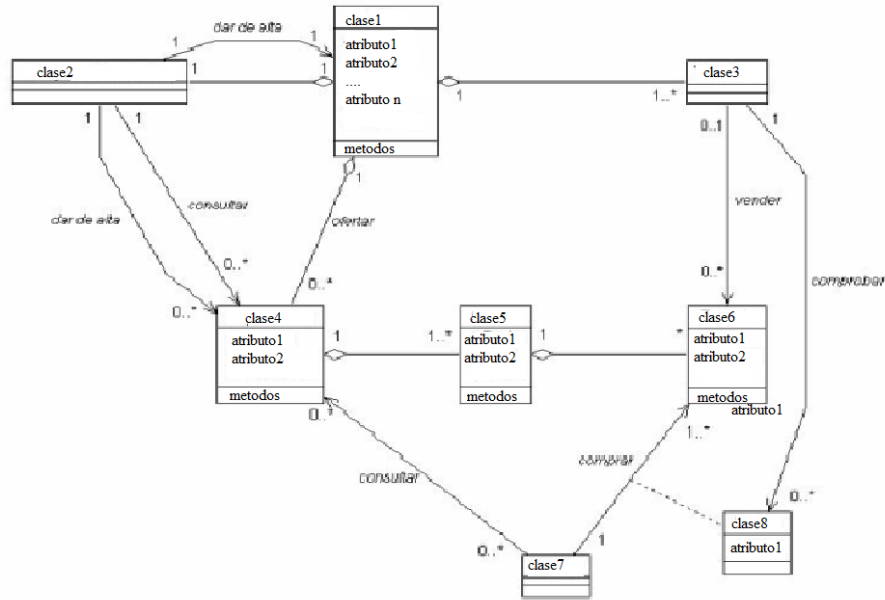


Figura 2.4. Diagrama de clases

Fuente: [MSO15]

### 2.5.3. Diagrama de secuencia

En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal. La figura 2.5., muestra la interacción de crear una secuencia.[MSO15]

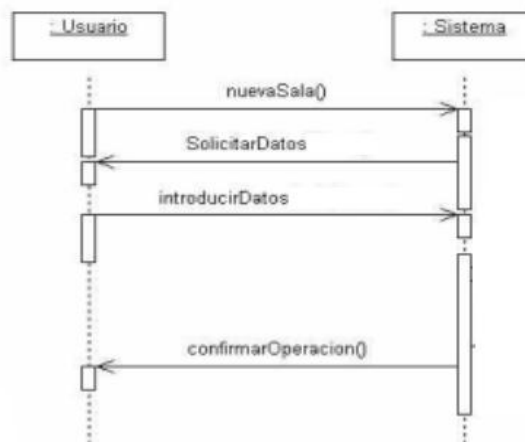


Figura 2.5. Diagrama de secuencia.

Fuente: [MSO15]

#### 2.5.4. Diagrama de estados

El comportamiento en tiempo real de cada clase que tiene comportamiento dinámico y significativo, se modela usando un Diagrama de Estado. El diagrama de actividad puede ser usado también aquí, esta vez como una extensión del diagrama de estado, para mostrar los detalles de las acciones llevadas a cabo por los objetos en respuesta a eventos internos. La figura 2.6., representa en una forma gráfica el diagrama de estado.[MSO15]

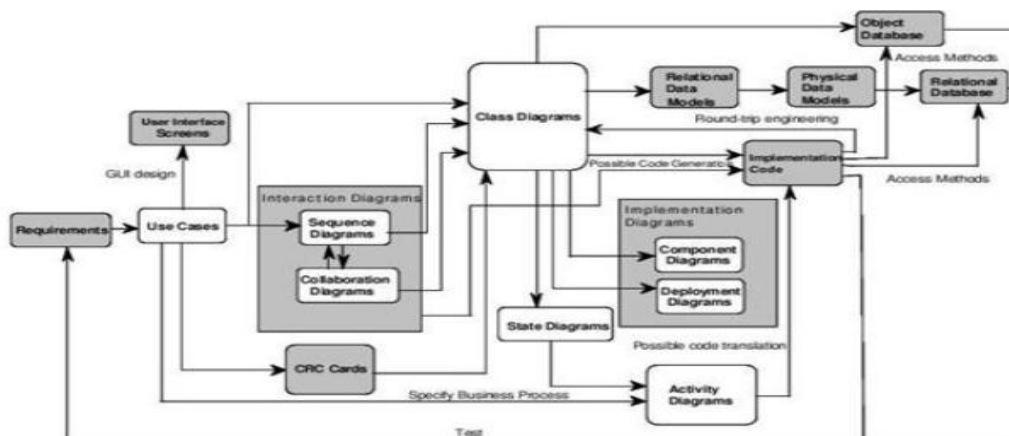


Figura 2.6. Diagrama de estado.

Fuente: [MSO15]

#### 2.5.5. Diagrama de paquetes

UML introduce la noción de un paquete como el ítem universal para agrupar elementos, permitiendo a los modeladores subdividir y categorizar sistemas. Los paquetes pueden ser usados en cualquier nivel, desde el nivel más alto, donde son usados para subdividir el sistema en dominios, hasta el nivel más bajo, donde son usados para agrupar casos de uso individuales, clases, o componentes. La figura 2.7., representa gráficamente los paquetes en una forma gráfica.[MSO15]

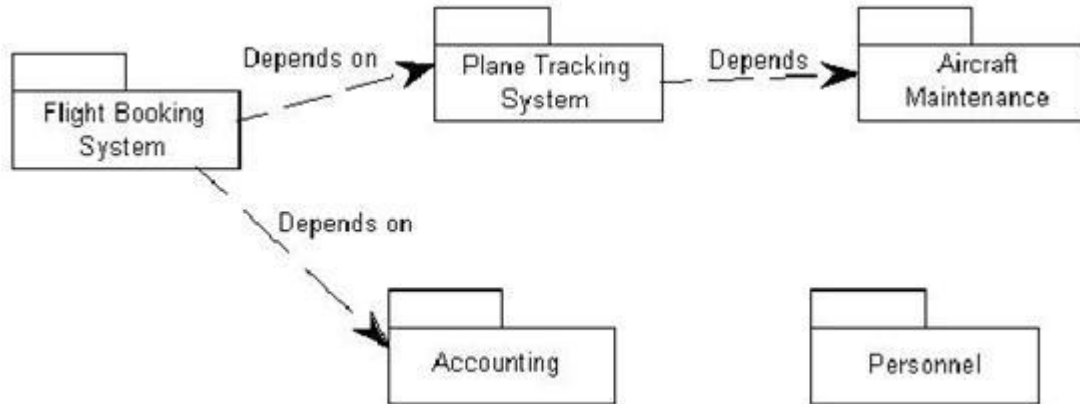


Figura 2.7. Diagrama de paquetes

Fuente: [MSO15]

## 2.6. Metodología de desarrollo Ágil SCRUM

SCRUM es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. SCRUM es una metodología ágil y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil incremental basada en iteraciones y revisiones.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve.

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto. Estas iteraciones son la base del desarrollo ágil, y SCRUM gestiona su evolución a través de reuniones breves, diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.[FXS08]

### 2.6.1. Fases de SCRUM

#### 2.6.1.1. Planes de lanzamiento

La planificación es una versión basada en la pila actual, junto con la estimación de coste y agenda. Si se trata de un nuevo sistema, esta fase abarca tanto la visión como el análisis.[FXS08]

### **2.6.1.2. Distribución, revisión y ajuste de los estándares del producto**

- Revisión de las iteraciones: Al finalizar cada iteración se lleva a cabo una revisión con todas las personas implicadas en el proyecto.
- Desarrollo incremental: Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.
- Desarrollo evolutivo: Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. Intentan predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no realista, porque las circunstancias obligarán a remodelarlo muchas veces.
- Auto-organización: Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles.
- Colaboración: Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. [FXS08]

### **2.6.1.3. Sprint**

Es el desarrollo de la funcionalidad de la nueva versión con respeto continuo a las variables de tiempo, requisitos, costo y competencia. La interacción con estas variables define el final de esta fase. El sistema va evolucionando a través de múltiples iteraciones de desarrollo o sprints.

- Control de la evolución del proyecto, Scrum controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas de la gestión ágil.
- Visión general del proceso, Scrum denomina “Sprint” a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El Sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental. Los elementos que conforman el desarrollo Scrum son:

Las reuniones:

<b>Planificación de Sprint</b>	Jornada de trabajo previa al inicio de cada sprint en la que se determina cual va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
<b>Reunión Diaria</b>	Breve revisión del equipo de trabajo realizado hasta la fecha y la previsión para el día siguiente.
<b>Revisión de Sprint</b>	Análisis y revisión del incremento generado.

Tabla 2.2. Tipos de reuniones de Scrum

Fuente: [FXS08]

Los elementos:

<b>Producto BackLog</b>	Lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo, evolucionando durante el desarrollo.
<b>Sprint BackLog</b>	Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
<b>Incremento</b>	Resultado de cada sprint.

Tabla 2.3. Elementos de Scrum

Fuente: [FXS08]

- Los roles, Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: propietario, equipo, gestor de Scrum o Scrum Manager y otros interesados. [FXS08]

#### 2.6.1.4.Revisión Sprint

Scrum es una ayuda para organizar a las personas y el flujo de trabajo, como lo pueden ser otras propuestas de formas de trabajo ágil.

Delegación de atribuciones (empowerment) al equipo para que pueda auto-organizarse y tomar las decisiones sobre el desarrollo.

Respeto entre las personas, los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.

Responsabilidad y auto-disciplina (no disciplina impuesta).

Trabajo centrado en el desarrollo de lo comprometido.

Información, transparencia y visibilidad del desarrollo del proyecto. [FXS08]



### **2.6.1.5.Cierre**

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión. [FXS08]

### **2.7. Arquitectura de software en el ciclo de desarrollo de Scrum**

La idea fundamental es adicionar un sprint inicial llamado “Sprint 0” al inicio del ciclo de desarrollo. El objetivo principal del arquitecto en el Sprint 0 es analizar y diseñar la generalidad del sistema, que satisfaga los requisitos y sea entendible por los miembros del equipo desde sus diferentes puntos de vista durante el desarrollo. Un punto clave, es reutilizar artefactos de software creados a partir de la arquitectura para ser más ágiles en el desarrollo de productos específicos.

En la Figura 2.8., se puede observar que el Sprint 0 comprende tres fases que fueron tomadas del ciclo de vida de entregas evolutivas. En el Sprint 0 se construye la arquitectura de forma iterativa mediante un análisis preliminar de los conductores arquitectónicos (requisitos funcionales, de calidad y del negocio), y de un estudio de factibilidad del proyecto. No se necesita tener todos los requisitos claros para comenzar la fase de diseño arquitectónico.

Para determinar los conductores arquitectónicos, se debe identificar los objetivos del negocio de más alta prioridad, que son pocos. Estos objetivos del negocio se convierten en los escenarios de calidad o casos de uso. De esta lista, se deben escoger los que tendrán un mayor impacto sobre la arquitectura. El diseño arquitectónico puede comenzar una vez que se encuentran definidos los conductores arquitectónicos. El proceso de análisis de requisitos será entonces influenciado por las preguntas generadas durante el diseño arquitectónico.

El resultado del Sprint 0 es un documento inicial que explica la arquitectura. El documento puede basarse en los pasos definidos por el método ADD (Attribute Driven Design). Este método ha sido probado exitosamente en proyectos anteriores. Con ADD se puede definir una arquitectura de software mediante un proceso de descomposición basado en los atributos de calidad del software.

El documento inicial de la arquitectura se debe evaluar con el fin de saber si la arquitectura cumple con los requisitos de calidad. Para realizar esta evaluación, se propone el método ATAM (Architecture Tradeoff Analysis Method). El ATAM revela cuán bien una arquitectura satisface los objetivos particulares de calidad y provee una aproximación de cómo los objetivos de calidad interactúan.[SSA12]

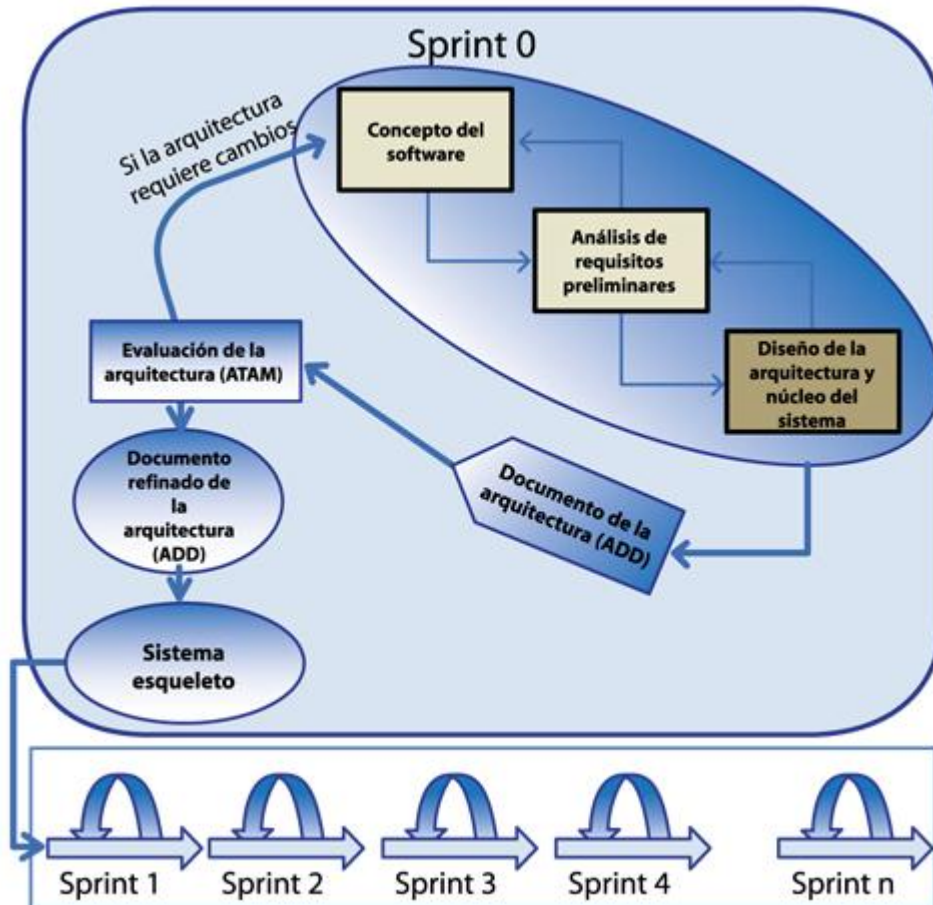


Figura 2.8. Propuesta de gestión de la arquitectura de software en Scrum.

Fuente: [SSA12]

## 2.8. Evaluación de Arquitecturas de Software con ATAM

El Architecture Tradeoff Analysis Method (ATAM), es una metodología para evaluar Arquitecturas de Software que principalmente evalúa la adecuación de la Arquitectura de Software definida con respecto a los atributos de calidad especificados para el sistema. Surge confluendo ideas y técnicas de tres áreas: la noción de estilos o patrones de arquitectura, el análisis de atributos de calidad y el método Software.

El corazón de ATAM consiste en la ejecución de nueve pasos que se dividen en cuatro grupos que su vez, se realizan en el tiempo en cuatro Fases diferenciadas. Estos cuatro grupos no se corresponden uno a uno con las Fases, sino que se realizan en las Fases 1 y 2, y se agrega una Fase 0 previa de preparación y una Fase 3 posterior de finalización del proyecto. Si bien la numeración de pasos sugiere linealidad, la ejecución de los mismos no necesariamente es un



proceso en cascada estricto, ya que se podrá volver a pasos anteriores o saltar hacia adelante a pasos posteriores, o inclusive iterar entre pasos según sea necesario. Los cuatro grupos en que se dividen los nueve pasos definidos consisten en un primer grupo de presentación, donde se intercambia información del sistema, un segundo grupo de investigación y análisis, donde se valoran los atributos de calidad claves uno a uno con las propuestas arquitectónicas, un tercer grupo de pruebas donde se revisan los resultados obtenidos contra las necesidades relevantes de los stakeholders (las partes interesadas), y un cuarto y último grupo donde se presentan los resultados del ATAM. En la Fase 0 se acuerdan tiempo, fechas, costos, esfuerzo y se forma el equipo de evaluación, en las Fases 1 y 2 se realiza la evaluación con ATAM siguiendo los nueve pasos especificados, y finalmente en la Fase 3 se realiza el informe final de la evaluación realizada, se recoge información para la medida y mejora del proceso y se actualizan los repositorios de productos generados. A continuación se describen las Fases 1 y 2 que constituyen la base de ATAM.[CDA15]

### **2.8.1. Fase 1:**

En la Fase 1 se realizan las actividades correspondientes a los grupos de presentación e investigación y análisis. El grupo de presentación se compone de tres pasos: 1 – Presentar el ATAM, 2 – Presentar las pautas del negocio y 3 – Presentar la Arquitectura. En el primer paso el equipo de ATAM presenta el método a los stakeholders explicando el proceso a seguir y el involucramiento y responsabilidad de cada uno en el proyecto. Se detallan los pasos a seguir, las técnicas a utilizar y los resultados a obtener. En el segundo paso un director de proyecto o gerente presenta el sistema desde el punto de vista del negocio, al equipo de ATAM y a los stakeholders, detallando principales funcionalidades restricciones y metas definidas para el sistema. En el tercer paso el Arquitecto presenta la Arquitectura de Software definida incluyendo por lo menos los estilos utilizados, otros sistemas con que se debe interactuar, restricciones técnicas de software como por ejemplo uso de sistema operativo. El segundo grupo de investigación y análisis se compone también de tres pasos: 4 – Identificar las propuestas arquitectónicas, 5 – Generar el árbol de utilidad de los atributos de calidad y 6 – Analizar las propuestas arquitectónicas. En el cuarto paso el equipo de ATAM le pide al Arquitecto que identifique las propuestas arquitectónicas o estilos de arquitectura utilizados, ya que éstos definirán las estructuras importantes

definidas para el sistema y las características implicadas. En el quinto paso se genera el árbol de utilidad donde principalmente el Arquitecto pero también los principales stakeholders, identifican, priorizan y refinan los requerimientos de atributos de calidad más importantes del sistema, según se mencionó previamente, identificando los escenarios en el árbol y su importancia en las dos dimensiones definidas. En el sexto paso se analizan las propuestas arquitectónicas según el árbol de utilidad generado, esto es, que tan adecuados son el uno para el otro, evaluando como cada propuesta arquitectónica influye en la obtención o no del atributo de calidad requerido, e identificando los riesgos, no riesgos, puntos de sensibilidad y concesión asociados a dicha evaluación.[CDA15]

### **2.8.2. Fase 2:**

En la Fase 2 se realizan las actividades incluidas en el tercer grupo de pruebas y el cuarto grupo de informes, que completan los tres pasos restantes. El grupo de pruebas se compone de dos pasos: 7 – Lluvia de ideas y 8 – Analizar las propuestas arquitectónicas y el grupo de Informes se compone de un solo paso: 9 – Presentar los resultados. En el séptimo paso se confirman e identifican nuevos escenarios según varios stakeholders involucrados, los que también se priorizan y se comparan con los identificados en el árbol de utilidad. Pueden suceder tres casos: el escenario ya está en el árbol de utilidad, no está pero encaja en alguna rama y se convierte en una nueva hoja, no está y no encaja en ninguna rama, lo que significa que no había sido considerado previamente. En el octavo paso se realiza lo mismo que en el sexto paso para el nuevo árbol de utilidad con todos los escenarios incluidos. Finalmente en el noveno paso el equipo de ATAM presenta los resultados a los stakeholders y entrega la documentación correspondiente a las salidas del ATAM: documento de propuestas arquitectónicas, conjunto de escenarios priorizados, conjunto de preguntas basadas en los atributos, árbol de utilidad, los riesgos descubiertos, los no riesgos documentados, los puntos de sensibilidad y de concesión encontrados, más la relación entre los riesgos encontrados y su impacto en las pautas del negocio definidas.[CDA15]

## **2.9.Requerimiento de software y hardware**

Para el desarrollo del siguiente proyecto se utilizaran, un conjunto de herramientas de software y hardware, de manera que estas herramientas coadyuven en el desarrollo del sistema en sus diferentes etapas, se hará uso de herramientas para el desarrollo de la programación JAVA y PostgreSQL.

### **2.9.1. Java**

Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa. Java permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios.

Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet. Java es un valor impagable para los desarrolladores, ya que permite: [JVE14]

- Escribir software en una plataforma y ejecutarla virtualmente en otra.
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles.
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más.
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización.
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, micro-controladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico.

[JVE14]

#### **2.9.1.1. Librerías Java**

Entre las librerías más utilizadas para el desarrollo del producto se encuentran:

- postgresql-9.1-903.jdbc3: Conexión al servidor Postgres.
- jfreechart-1.0.13: Generación de gráficas.
- jcalendar-1.3.3: Obtener el calendario de forma dinámica.
- Jxl: Exportación a hojas Excel.
- poi-3.9-20121203: Exportación a documentos Word docx.

## **2.9.2. PostgreSQL**

PostgreSQL es un potente sistema de base de datos, de código abierto objeto-relacional. Cuenta con una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección.

Es compatible con los juegos de caracteres internacionales, codificación de caracteres multi-byte, Unicode, y es consciente de la configuración regional de clasificar, mayúsculas y minúsculas, y el formato. Es altamente escalable tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios simultáneos que puede acomodar. [PGS12]

La seguridad de la base de datos esta implementada en varios niveles:

Protección de los ficheros de la base de datos. Todos los ficheros almacenados en la base de datos están protegidos contra escritura por cualquier cuenta que no sea la del súper usuario de Postgres.

Las conexiones de los clientes se pueden restringir por dirección IP y/o por nombre de usuario mediante el fichero pg\_hba.conf situado en PG\_DATA.

A cada usuario de Postgres se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no hayan creado.

Los usuarios pueden ser incluidos en grupos, y el acceso a las tablas puede restringirse en base a esos grupos.

La autenticación de usuarios es el proceso mediante el cual el servidor de la base de datos y el post-master se aseguran de que el usuario que está solicitando acceso a la base de datos es en realidad quien dice ser. [SBD14]

## **2.10.Prueba**

### **2.10.1. Prueba Unitaria**

Un test unitario (Unit Test) es un trozo de código desarrollado con el único objetivo de verificar que una rutina o función de nuestro código está funcionando según esperamos.[PURR10]

Debido al gran éxito que esta metodología está teniendo en el mundo de la ingeniería del software, son muchas y variadas las herramientas que se han ido desarrollando

hasta el día de hoy. Algunas han surgido en forma de Frameworks, otras en formato plugin bajo el control de algún browser o incluso como librerías para su uso en plataformas de programación.

Entre los que podemos citar a:

- JUnit: Entorno de pruebas para Java creado por Erich Gamma y Kent Beck.
- HttpUnit: viene a ser un cliente web programable. Se puede usar de forma aislada o como complemento del framework Junit.
- DBUnit: en realidad es una extensión de JUnit que tiene como ventaja el poder tener en cuenta la existencia de una base de datos a la hora de realizar las pruebas de los test.
- SimpleTest: Entorno de pruebas para aplicaciones realizadas en PHP.
- PHPUnit: Framework para realizar pruebas unitarias en PHP.
- CPPUnit: Versión del Framework para lenguajes C/C++.

La mayoría de las herramientas tienen assert, que son afirmaciones que son usada para comprobar suposiciones en el programa, colocada donde el desarrollador considera que su enunciado es siempre verdadero.

Esto da lugar a que existan afirmaciones y por lo tanto condiciones antes(pre-condiciones) y después (post-condiciones) de la ejecución de determinadas líneas de código, lo que da lugar a las pruebas unitarias. [PURR10]

## **2.11.Calidad del Software**

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

La calidad es la aptitud de un producto o servicio para satisfacer las necesidades del usuario.

Es la cualidad de todos los productos, no solamente de equipos sino también de programas.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un



aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta. [CDS12]

La calidad del software se define como: concordancia con los requisitos de rendimiento explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.[PRESS]

La calidad del software consiste en aquellos procedimientos, técnicas e instrumentos aplicados por entes capacitados para garantizar que un producto cumpla o supere un nivel mínimo aceptable para su comercialización, si es que así se lo ha planteado, lo que hasta el momento no se tiene estándares del software, no es lo mismo que las pruebas del sistema, estos son implícitos al momento de hacer las pruebas de integración final del sistema.[PRESS]

### 2.11.1. Funcionalidad

El punto función es una métrica orientada a la función. Es una medida indirecta del software y del proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa. Los puntos de función se calculan llenando los datos en la figura para el cálculo se determinan cinco características de dominios de información:

Parámetros de medición	Cuenta	Factor de ponderación					
		Simple	Medio	Complejo			
Número de entradas de usuario		x	5	4	3	=	
Número de salidas de usuario		x	6	9	3	=	
Número de peticiones de usuario		x	1	13	0	=	
Número de archivos		x	1	9	5	=	
Número de interfaces externas		x	6	4	0	=	
Cuenta total	→						

Tabla 2.4. Factor de ponderación

Fuente: [Elaboración propia]

- El número de entradas de usuarios. Se cuenta cada entrada de usuario que proporciona al software de diferentes datos orientados a la aplicación. Las entradas deben diferenciar de las peticiones, estas pueden ser el ingreso de datos o control de flujo de

la información, y otros datos importantes. Dichas entradas se llevan a cabo mediante las siguientes interfaces.

- Pantalla de inicio del sistema, pantalla de control de permisos, grupos y usuarios, Pantalla de control de registros de producción.
- Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. Las salidas de usuario representan los informes, pantallas, mensajes de error. En nuestro ámbito tenemos como salidas: los mensajes de error, mostrar los registros de producción, y otras interfaces que se desarrollan en el software.
- Número de peticiones de usuario. Una petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta inmediata del software en forma de salida interactiva. Se cuenta por separado cada petición, en este caso una petición viene a ser: la consulta de rendimientos, de pH, de horas pasteurizadas, horas de fabricación y otras consultas desarrolladas en el software.
- Número de archivos. Se cuenta cada archivo lógico maestro, es decir un grupo lógico de datos que pueden ser parte de una gran base de datos o un archivo independiente. Se cuenta con un archivo de configuración del sistema, el cual permite hacer el enlace con el servidor Postgres, además de configurar los clientes.
- Numero de interfaces externas. Se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir información a otro sistema.

### **2.11.2. Confiabilidad**

Para determinar la confiabilidad del sistema se especifica el instante en el que comienza a funcionar determinado por  $t_0=0$ . A partir de este momento se observa el trabajo del sistema hasta que se introduzca una falla en el instante T que se va aproximando a una variable aleatoria continua, que nos determina la confiabilidad en términos probabilísticos. Entonces se tiene las siguientes probabilidades:

$P(T \leq t) = F(t)$  ... (1) Probabilidad de fallas

$P(T > t) = 1 - F(t)$  ... (2) Probabilidad de trabajo sin fallas

Se calcula estas probabilidades mediante la distribución exponencial. Donde, punto función, resultado obtenido en el cálculo realizado en un punto anterior a este.

El F(t) está dada por:  $F(t) = \text{Punto función} * e^{(-\lambda * t)}$

### 2.11.3. Mantenimiento

Se calcula el índice de madurez del software (IMS), estableciendo los cambios que ocurrieron con la versión del producto.

MT: Nro. De módulos en la versión actual.

Fc: Nro. De módulos en la versión actual que se han cambiado.

Fa: Nro. De módulos en la versión actual que se han añadido.

Fe: Nro. De módulos en la versión actual que se han eliminado.

El IMS está dada por:  $IMS = (MT - (Fc + Fa + Fe)) / MT$

### 2.11.4. Certificación del software

Consecuencia de un proceso que es asegurar la calidad pero nunca es el objetivo final. La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización ISO 9000.

La normativa ISO 9000 pone a disposición de un auditor o certificador los procesos internos, de forma que este indique si cumple o no la normativa al 100%, audita el sistema. Si los resultados son positivos se emite la certificación y cada cierto tiempo se tiene que renovar. La certificación es costosa, a consecuencia de costes que ocasionan la lejanía y el tiempo de duración de proceso (aprox. 6 meses). Se certifica la empresa y la metodología para el desarrollo de la aplicación. [CDS12]

### 2.11.5. Medición del software

En el software lo que se mide son atributos propios del mismo, se descompone un atributo general en otros más simples de medir, a veces se mide bien o mal ya que la descomposición del atributo genérico de calidad en otros sub-atributos se torna irreal, se mide con datos estadísticos no avalados, es imposible decir que la medición se hace en forma correcta. [CAFU14]

El concepto de medida va de más a menos, va de lo general a lo concreto y lo concreto es asociado a la métrica, cuya combinación te daría el nivel de calidad o seguridad de tu producto. Las ciencias bien estructuradas se basan en medidas bien hechas, se basan en la matemática. Los tipos de medidas son:

- Número de errores durante un periodo determinado.



- Fallo en la codificación o diseño de un sistema que causa que el programa no funcione correctamente o falle.
- Tamaño de un producto informático (líneas de código)
- Métrica de punto función (IBM): relaciona funcionalidades que ofrece estimación de costes y esfuerzos. [CAFU14]

## 2.12. Estimación de costos

### 2.12.1. COCOMO

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés Constructive Cost Model) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. [COSC95]

Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente.

#### - Modelos de estimación

Las ecuaciones que se utilizan son:

$$E = a(Kl)^b * m(X), \text{ en persona-mes}$$

$$Tdev = c(E)^d, \text{ en meses}$$

$$P = E/Tdev, \text{ en personas}$$

Dónde:

E es el esfuerzo requerido por el proyecto, en persona-mes

Tdev es el tiempo requerido por el proyecto, en meses

P es el número de personas requerido por el proyecto

a, b, c y d son constantes con valores definidos en una tabla, según cada sub-modelo

Kl es la cantidad de líneas de código, en miles.

m(X) Es un multiplicador que depende de 15 atributos.

[COSC95]

A la vez, cada sub-modelo también se divide en modos que representan el tipo de proyecto, y puede ser:

**Modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).

**Modo semilibre o semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

**Modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

[COSC95]

### Modelo básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla (ver tabla 2.5.) de constantes para calcular distintos aspectos de costes:

MODO	a	b	c	D
Orgánico	2.40	1.05	2.50	0.38
Semi - Orgánico	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Tabla 2.5. Constantes para calcular distintos aspectos de costes.

Fuente: [COSC95]

Estos valores son para las fórmulas:

Personas necesarias por mes para llevar adelante el proyecto (**MM**) =  $a \cdot (Kl)^b$

Tiempo de desarrollo del proyecto (**TDEV**) =  $c \cdot (MM)^d$

Personas necesarias para realizar el proyecto (**CosteH**) =  $MM/TDEV$

Costo total del proyecto (**CosteM**) =  $CosteH \cdot \text{Salario medio entre los programadores y analistas.}$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal.

Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.[COSC95]

### **Modelo intermedio**

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son (ver tabla 2.6.):

<b>MODO</b>	<b>A</b>	<b>b</b>
<b>Orgánico</b>	3.20	1.05
<b>Semi – Orgánico</b>	3.00	1.12
<b>Empotrado</b>	2.80	1.20

Tabla 2.6. Constantes a reemplazar en la fórmula.

Fuente: [COSC95]

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño, mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semi-libre con respecto al efecto multiplicador de los atributos de coste.[COSC95]

### **Modelo Detallado**

Presenta principalmente dos mejoras respecto al anterior:

Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, y otros. Tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra. [COSC95]

De esta manera haciendo el uso del software COCOMO y utilizándose las fórmulas mencionadas anteriormente, se logra estimar los costos, los recursos humanos y el tiempo, el software se encuentra publicado en la WEB y es de uso libre. [COSC95]

### **2.13.Seguridad del sistema**

Siendo que la información es el activo más importante que posee cualquier sistema, deben existir técnicas más allá de la seguridad física, que la aseguren.

Estas técnicas las brinda la seguridad lógica. [SSI12]

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y solo se permita acceder a ellos a las personas autorizadas para hacerlo. Para tal efecto se consideran los siguientes aspectos:

### **2.13.1. Controles de acceso**

Estos controles pueden implementarse en el sistema operativo, sobre los sistemas de aplicación, en base de datos, en un paquete específico de seguridad y en cualquier otro utilitario.

Constituyen una importante ayuda para proteger al sistema operativo de la red, al sistema de aplicación y demás software de la utilización o modificaciones no autorizadas; para mantener la integridad de la información (restringiendo la cantidad de usuarios y procesos con acceso permitido) y para resguardar la información confidencial de accesos no autorizados. [SSI12]

### **2.13.2. Identificación y Autenticación**

Es la primera línea de defensa para la mayoría de los sistemas computarizados, permitiendo prevenir el ingreso de personas no autorizadas. Es la base para la mayor parte de los controles de acceso y para el seguimiento de las actividades de los usuarios.

Se denomina identificación al momento en que el usuario se da a conocer en el sistema; y autenticación a la verificación que realiza el sistema sobre esta identificación. [SSI12]

### **2.13.3. Roles**

El acceso a la información también puede controlarse a través de la función o rol de usuario que requiere dicho acceso. Algunos ejemplos de roles serían los siguientes:

- Programador, líder de proyecto.
- Gerente de un área, Administrador del sistema.

En este caso los derechos de acceso pueden agruparse de acuerdo con el rol de los usuarios. [SSI12]

**CAPITULO III**  
**MARCO APLICATIVO**

**3.1. Introducción**

Según la ingeniería de software, en el diseño y desarrollo de todo producto software existen una serie de actividades que deben realizarse en un orden determinado y que abarcan no solo su producción, sino también su explotación y mantenimiento, esto se denomina ciclo de vida o proceso de desarrollo del Software.[PRESS]

El presente proyecto enfoca la problemática que existe en el área de producción de la empresa Flor de Leche SRL., los procesos manuales que existen, el cálculo del rendimiento, generación de gráficas, el histórico de rendimientos, en donde se desarrolla un sistema para el control de la elaboración de productos lácteos, a fin de mejorar la calidad de las funciones del encargado de Gerencia de producción. Basada en la metodología SCRUM realizamos el inicio de nuestro proyecto con los planes de lanzamiento en el sprint 0.

**3.2. Planes de lanzamiento**

Planificación del Sprint, de acuerdo a la (figura 2.8. del capítulo II) ,propuesta de la gestión de la arquitectura de software en Scrum. Ver tabla 3.1.

Sprint	Fases	Agenda
Sprint 0	Concepto del Software	15-12-14 AL 29-12-14
	Análisis de requisitos preliminares	
	Diseño de la arquitectura con UML	
	Evaluación de la arquitectura con ATAM	30-12-14 AL 31-12-14
	Documento refinado de la arquitectura	02-01-15 AL 03-01-15
	Sistema esqueleto o prototipo	05-01-15 AL 07-01-15

Tabla 3.1. Planes de lanzamiento del Sprint 0.

Fuente: [Elaboración propia]

**3.3. Sprint 0**

Gerencia de producción (GPR), responsable de la producción de productos lácteos requiere de un sistema de información que permita cumplir varios requerimientos los cuales colaboren con

la información que provee, GPR tiene el objetivo de sistematizar de mejor manera la información que lleva día a día, para ello se realiza una ingeniería de requerimientos.

### 3.3.1. Análisis de requisitos preliminares

Los requerimientos funcionales que describen las interacciones entre el sistema y el entorno son representados en la tabla 3.2.

<b>Código</b>	<b>Requerimiento</b>
<b>RF1</b>	Diseñar la base de datos, que permita almacenar la elaboración de productos lácteos.
<b>RF2</b>	Se debe manejar roles, usuarios, grupos de usuarios y permisos.
<b>RF3</b>	La inserción, modificación, eliminación de la fabricación debe ser flexible y debe tomarse en cuenta (La fecha de producción, el proceso, la recepción de leche, el producto, los moldes, los kilos finales, las horas de pasteurización, las horas de fabricación, el responsable de fabricación, el pH, la altura en cm).
<b>RF4</b>	La inserción, modificación y eliminación de tipos de productos se debe tomar en cuenta la siguiente información (descripción, y la unidad).
<b>RF5</b>	El registro de productos debe tomar en cuenta la siguiente información (descripción, relación-molde, y llevar una relación con los tipos de producto).
<b>RF6</b>	El sistema debe calcular de manera automática el rendimiento.
<b>RF7</b>	Toda la información debe tener la posibilidad de ser exportada a una planilla Excel.
<b>RF8</b>	Los reportes de rendimiento para poder ser editados se debe exportar a un formato WORD.
<b>RF9</b>	La información para los reportes de fabricación debe tomarse en cuenta de acuerdo a las fechas de selección.
<b>RF10</b>	Un reporte debe indicar los rendimientos de la fecha a la fecha de cada producto.
<b>RF11</b>	La graficas de rendimientos deben ser presentadas en dos formas: presentación de la gráfica y la presentación en el reporte.
<b>RF12</b>	Un reporte debe indicar los kilos y moldes producidos mensualmente y anualmente, por producto.
<b>RF13</b>	Un reporte debe indicar la desviación estándar por Responsable de fabricación.



<b>RF14</b>	Un reporte debe indicar los niveles de pH de la fecha a la fecha por producto.
<b>RF15</b>	La información de los niveles de pH se debe poder presentar en una gráfica y en los reportes.
<b>RF16</b>	Un reporte debe indicar el rendimiento promedio por mes, año, por producto.
<b>RF17</b>	Un reporte debe indicar el pH promedio por mes, año, por producto.
<b>RF18</b>	Un reporte debe indicar la desviación estándar por producto
<b>RF19</b>	Un reporte debe indicar el tiempo de fabricación, por semana, mes, año.

Tabla 3.2. Lista de requerimientos funcionales del sistema.

Fuente:[Requerimiento del cliente]

Los requerimientos de hardware que permiten que el sistema funcione correctamente se describen en la Tabla 3.3.

<b>Código</b>	<b>Requerimiento</b>
<b>RH1</b>	PC de escritorio o Laptop
<b>RH2</b>	Memoria 512 MB o superior
<b>RH3</b>	Espacio de disco duro 10 GB o superior
<b>RH4</b>	Procesador Pentium 4 o superior de 32 b
<b>RH5</b>	Pantalla, teclado, mouse, impresora compatible con el RH1

Tabla 3.3. Requerimientos de Hardware

Fuente: [Elaboración propia]

Los requerimientos de software que permiten que el sistema funcione correctamente se describen en la Tabla 3.4.

<b>Código</b>	<b>Requerimiento</b>
<b>RS1</b>	SO Windows XP o superior de 32 b
<b>RS2</b>	Microsoft Office 2007 o superior
<b>RS3</b>	Java 1.7. o superior
<b>RS4</b>	Winrar o Winzip

Tabla 3.4. Requerimientos de Software

Fuente: [Elaboración propia]

### 3.3.2. Diseño de la arquitectura

De acuerdo a los requerimientos funcionales se inicia con el diseño utilizando la metodología UML.

#### 3.3.2.1. Diagrama de casos de uso

La figura 3.1., representa la interacción de los actores con el sistema.

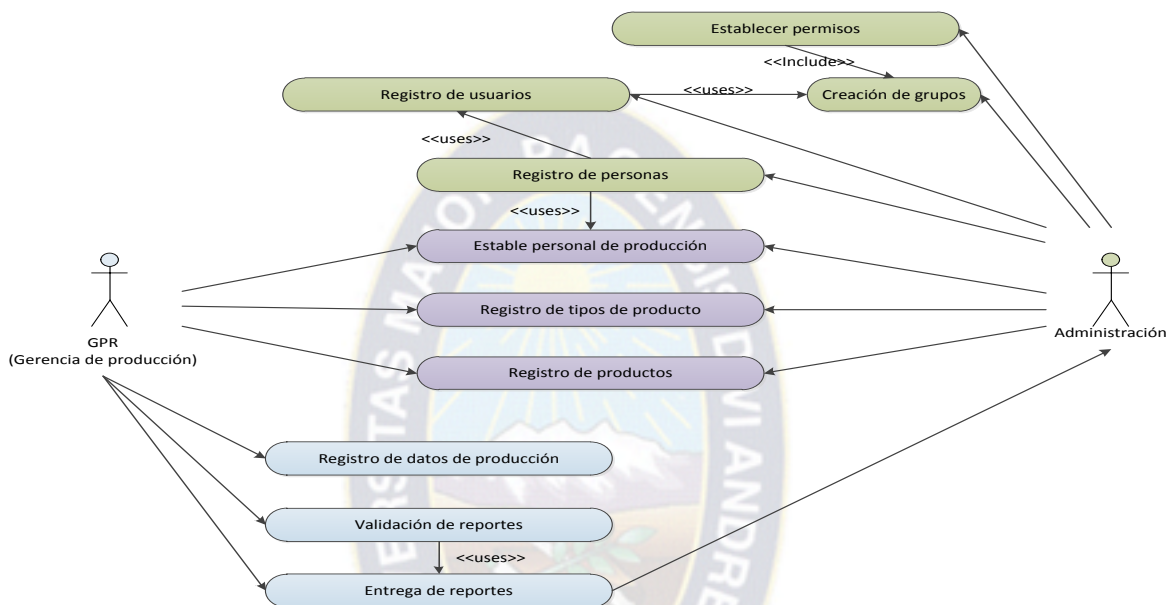


Figura 3.1. Modelado de casos de uso.

Fuente: [Elaboración propia]

De acuerdo a los casos de uso se detallan los más relevantes. Registrar datos de producción, (ver tabla 3.5.), validación de reportes (ver tabla 3.6.), establecer personal de producción (ver tabla 3.7.) y el resto de las descripciones de casos de uso se encuentra en el Anexo B.

<b>Nombre:</b>	<b>Registrar datos de producción</b>	
<b>Actor:</b>	<b>GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de registrar la información más importante, la cual hace que el sistema funcione correctamente.</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica la duplicidad de la información	
	Registrar los nuevos productos	Hacer validaciones de números, fechas,



	procesados.	y del producto escogido.
	Se emite los códigos de los nuevos registros procesados	El sistema emite los códigos de los productos procesados.
Pre-condición	Datos correctamente insertados en la tabla.	
Post-condición	Reportes confiables.	
Presunción	Las base de datos cuenta con la tabla de fabricación	

Tabla 3.5. Registrar datos de producción

Fuente: [Elaboración propia]

La descripción de otro caso de uso, muy relevante es la validación de reportes, ver tabla 3.6.

<b>Nombre:</b>	<b>Validación de reportes</b>	
<b>Actor:</b>	<b>GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de observar la información sobre los reportes generados y verificar si son correctos.</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Seleccionar la información, aplicando los filtros	Mostrar la información que requiere el usuario.
	Observar información y prever información errónea.	
	Emitir los reportes seleccionados, exportando a WORD, Excel o imprimiendo directamente.	Generación de reportes.
Pre-condición	Tener información seleccionada	
Post-condición	Reportes listos para la presentación	
Presunción	El sistema cuenta con los drivers necesarios para la exportación de datos.	

Tabla 3.6. Validación de reportes.

Fuente: [Elaboración propia]

La siguiente tabla describe el proceso de convertir a un empleado en parte del personal de producción (ver tabla 3.7.).

<b>Nombre:</b>	<b>Establecer personal de producción</b>	
<b>Actor:</b>	<b>Administración/GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de volver a los empleados parte del personal de producción.</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los datos necesarios para asignarlo parte del personal de producción	
	Registrarlo como empleado	Guarda la información en la BD y permite que aparezca en los responsables de fabricación.
<b>Pre-condición</b>	Existe el empleado en la BD.	
<b>Post-condición</b>	Ahora es responsable de fabricación.	
<b>Presunción</b>	Las base de datos cuenta con la tabla de personal de producción	

Tabla 3.7. Establecer personal de producción.

Fuente: [Elaboración propia]

### 3.3.2.2. Diagrama de clases

La figura 3.2., representa las clases encontradas y sus relaciones, así mismo tomar en cuenta que cada clase tiene respectivamente y por defecto las operaciones de Alta, Baja y Modificar. Además de incluir las funciones agregadas en la Base de datos las cuales permiten hallar la desviación estándar del pH, la desviación estándar del rendimiento, el promedio de pH, el promedio del rendimiento, y mediante estas funciones se pueden hallar los límites de advertencia y el pH o rendimiento esperado.

- devestph
- devestrendimiento
- promedioph
- promediorendimiento

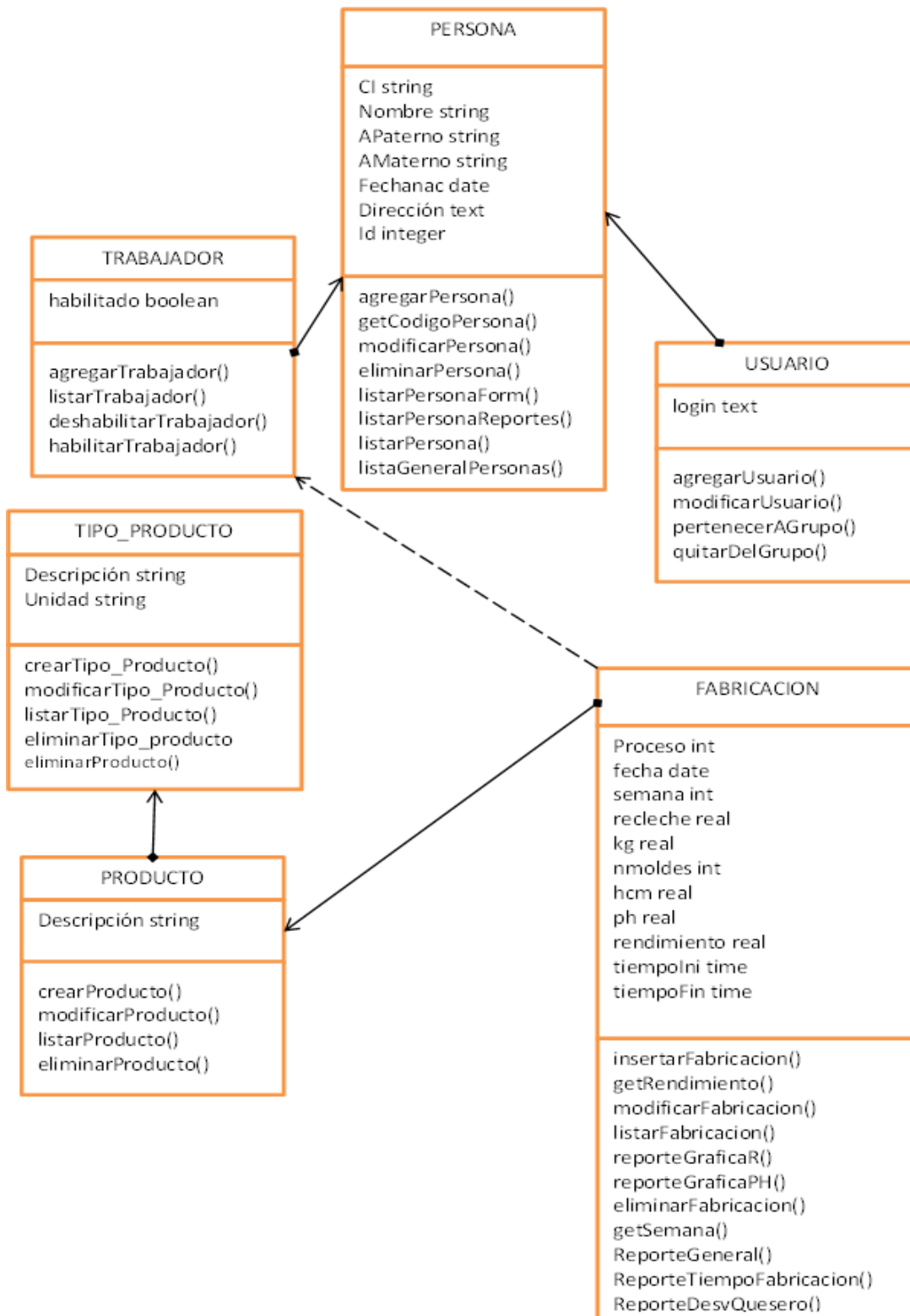


Figura 3.2. Diagrama de clases.

Fuente: [Elaboración propia]

La figura 3.3., representa el diagrama entidad – relación, de esta manera podemos interpretar las dependencias entre las entidades.

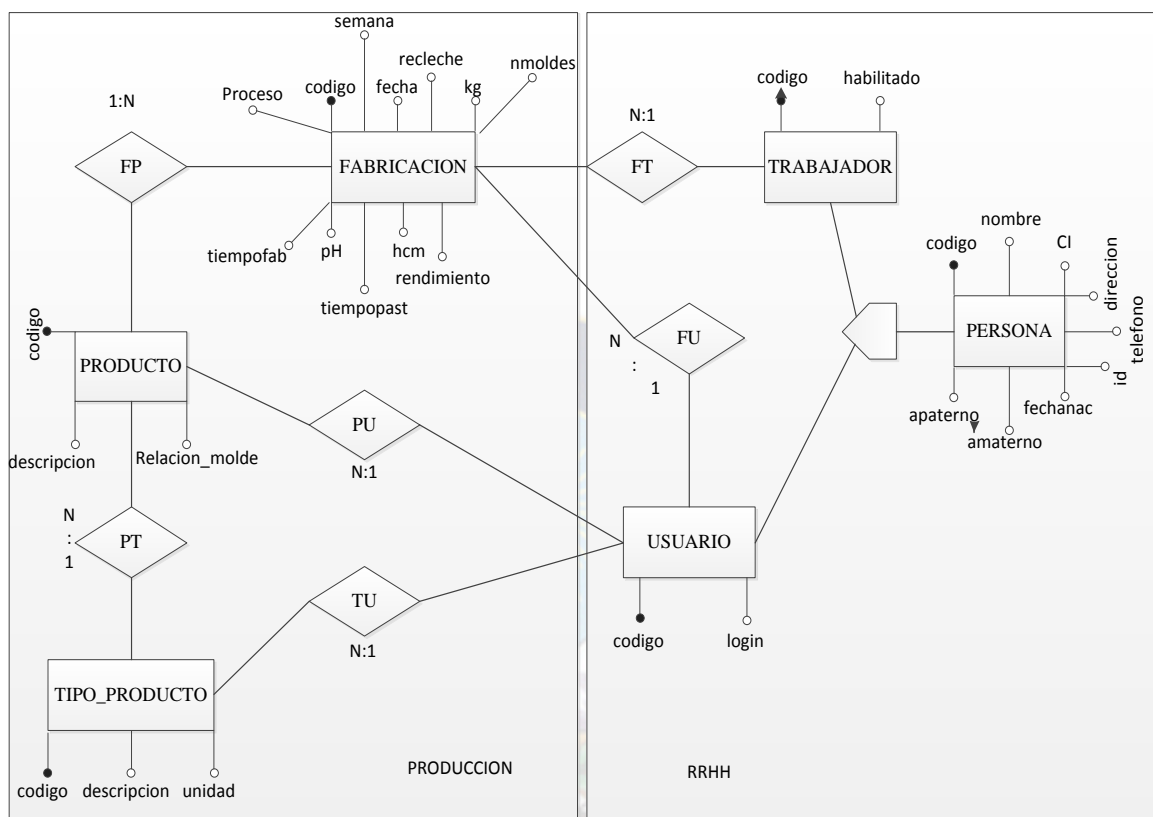


Figura 3.3. Diagrama Entidad-Relación.

Fuente: [Elaboración propia]

Debido a que el diagrama de Clase presenta un mecanismo de implementación neutral para modelar los aspectos de almacenado de datos del sistema. Las clases persistentes, sus atributos y sus relaciones pueden ser implementados directamente en un diagrama entidad relación (ver Figurar 3.3.), el diagrama de clase se puede usar para modelar la estructura lógica de la base de datos, independientemente de si es orientada a objetos o relacional, con las clases representando tablas y atributos de clase representado en columnas. Si una base de datos relacional es el método de implementación escogido, entonces el diagrama de clase puede ser referenciado a un diagrama entidad relación. Las clases persistentes y sus atributos hacen referencia directamente a las entidades lógicas y a sus atributos.

Entonces para empezar el proceso se determina si el modelo entidad relacional encaja, y que atributos son claves primarias, claves secundarias y claves externas basadas en relaciones con

otras entidades. La idea es construir un modelo lógico que sea conforme a las reglas de normalización de datos.

La figura 3.4., es la representación del modelo Relacional, la cual permite representar el diseño de la Base de Datos.

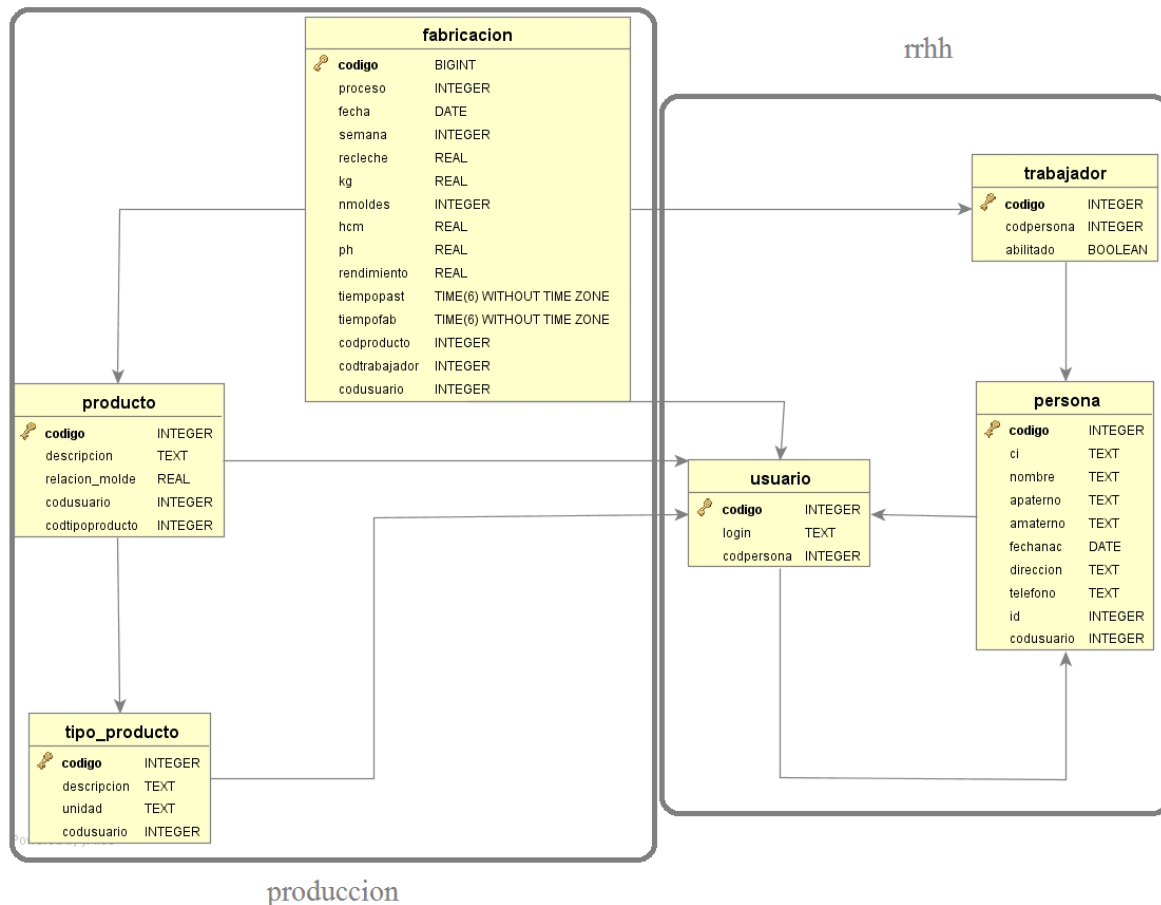


Figura 3.4. Diagrama Relacional.

Fuente: [Elaboración propia]

### 3.3.2.3. Diagrama de secuencia

De acuerdo a los diagramas de secuencia se explica la interacción y el seguimiento de cada uno de los procesos descritos en los casos de uso. Se presentan los diagramas de secuencias más relevantes, el diagrama de secuencias del registro de datos de producción (ver figura 3.5.), el diagrama de secuencias de la validación de reportes (ver figura 3.6), diagrama de secuencia del registro de productos (ver figura 3.7.), además se adjunta todos los diagramas de secuencia en el Anexo C.

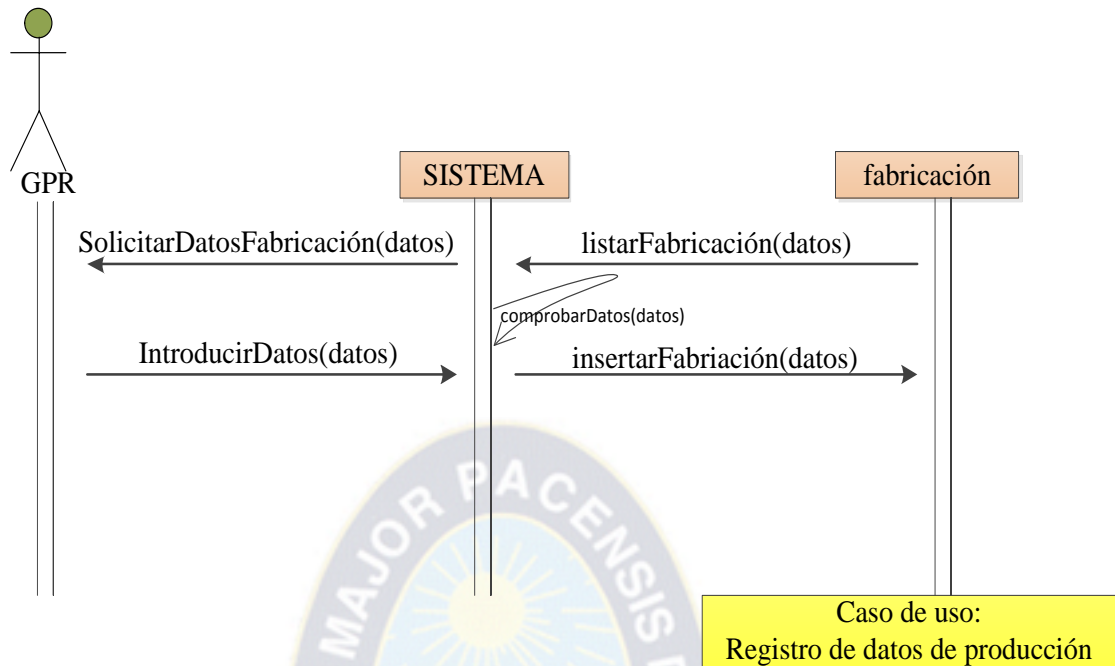


Figura 3.5. Diagrama de registro de datos de producción.

Fuente: [Elaboración propia]

Figura 3.6., representa el diagrama para la validación de reportes.

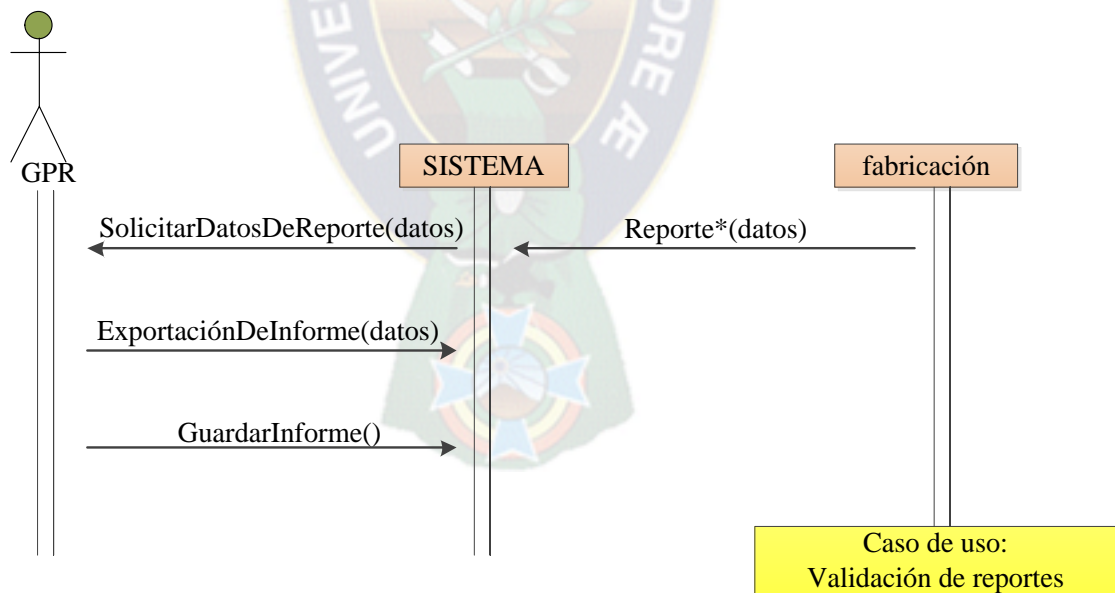


Figura 3.6. Diagrama de validación de reportes.

Fuente: [Elaboración propia]

Figura 3.7., representa el diagrama para el registro de productos.

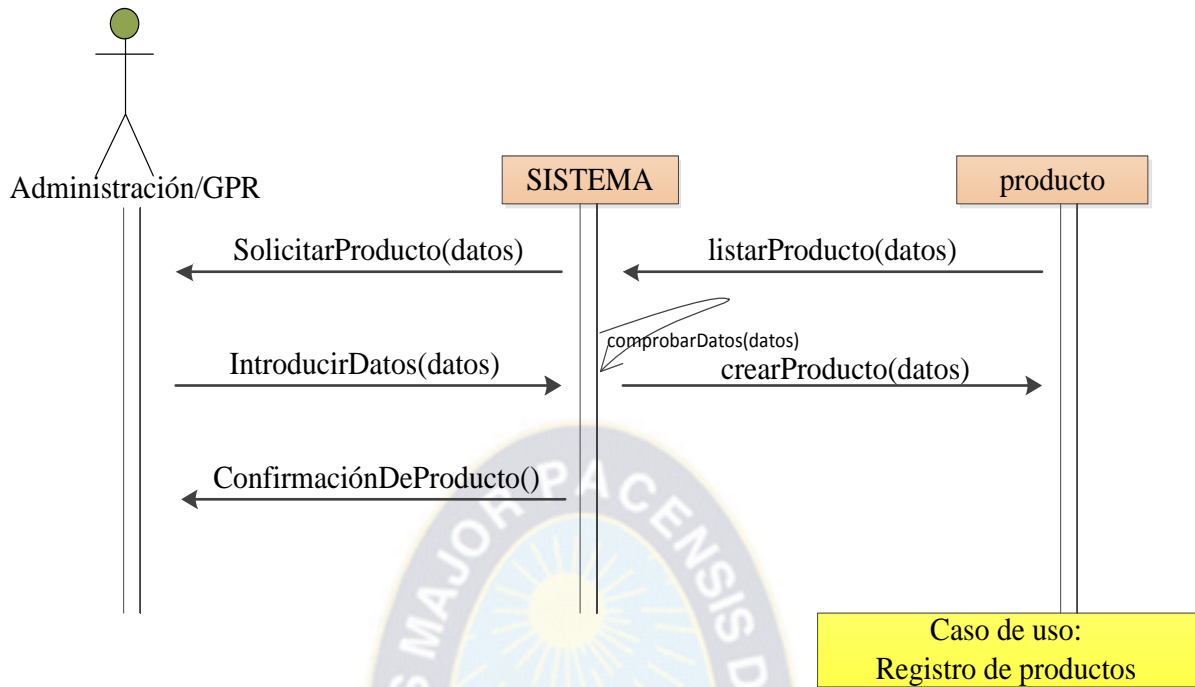


Figura 3.7. Diagrama del registro de productos.

Fuente: [Elaboración propia]

### 3.3.2.4. Diagrama de estados

La figura 3.8., representa la interacción de los estados entre el sistema y los actores.

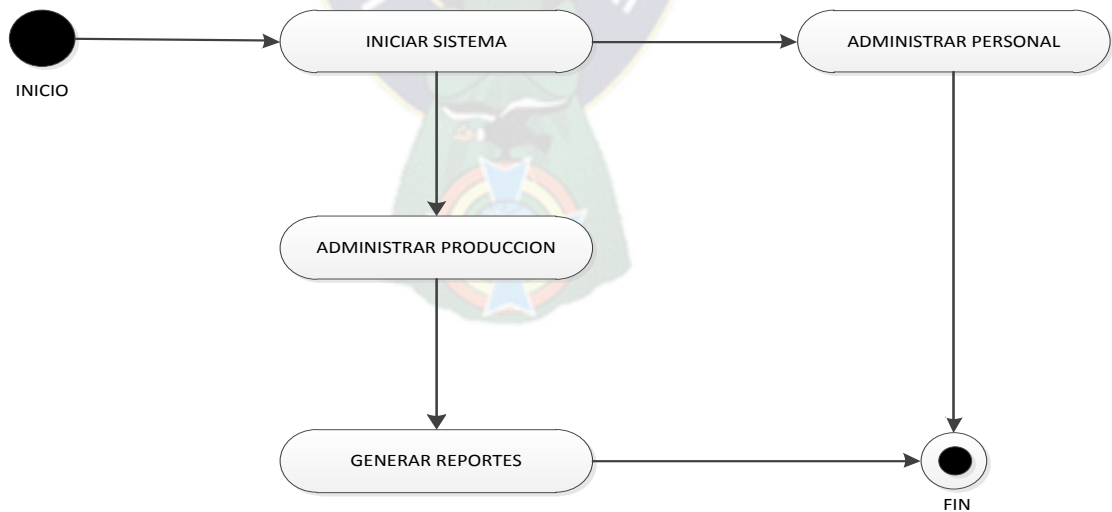


Figura 3.8. Diagrama de estados.

Fuente: [Elaboración propia]



### 3.3.2.5. Diagrama de paquetes

Dentro del análisis del sistema está la categorización. Esto es, los diferentes elementos de los modelos de análisis que se clasifican de una manera que se empaqueten como una agrupación, la figura 3.9., identifica como se agrupa el sistema.

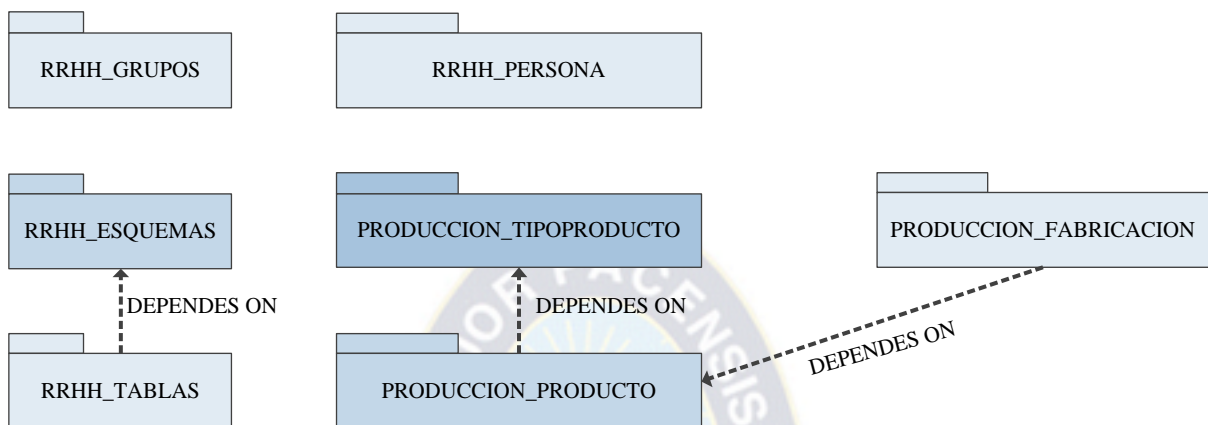


Figura 3.9. Diagrama de paquetes.

Fuente: [Elaboración propia]

### 3.3.3. Evaluación de la arquitectura

#### Problemas detectados

##### Problema 1

No se encontró un mapeo de todos los requerimientos al Modelo de Diseño.

##### Sugerencia de corrección

Realizar todos los diseños necesarios para cubrir los requerimientos, para que haya trazabilidad entre los documentos.

##### Problema 2

Aunque se toma en cuenta al armar la arquitectura, no se encontró una estrategia descrita para solucionar problemas de seguridad, como ser la integridad de los datos a manejar.

##### Sugerencia de corrección

Incluir en la documentación las estrategias de mitigaciones de este riesgo. (Utilización de pgadmin).

##### Problema 3

Falta definir como se hará el manejo de errores.

##### Sugerencia de corrección

Incluir en el diagrama de secuencias como se realizará esta validación.

### **Evaluación**

La arquitectura se puede implementar, cubre las propiedades de calidad (poniendo énfasis en la extensibilidad tanto de funcionalidades como en la de perfiles, ya que este fue el principal tema cuando se negoció el alcance) y se tomaron en cuenta posibles riesgos.

### **Estado actual del Producto**

Tanto el documento de Descripción de la Arquitectura como el Modelo de Diseño deben completarse en base a los casos de uso más relevantes del proyecto. Se deben actualizar y re entregar los documentos.

### **Acciones a tomar**

El documento será actualizado tomando en cuenta lo discutido en la reunión ya que todos los integrantes aportaron ideas. Será re-entregado.

### **Próxima Revisión del Producto**

Se hará una próxima revisión antes de comenzar con la fase de construcción.

## **3.4.Sistema esqueleto o prototipo**

Se realiza un prototipo base la cual permite establecer los principales formularios.

Ver figura 3.10., y figura 3.11.

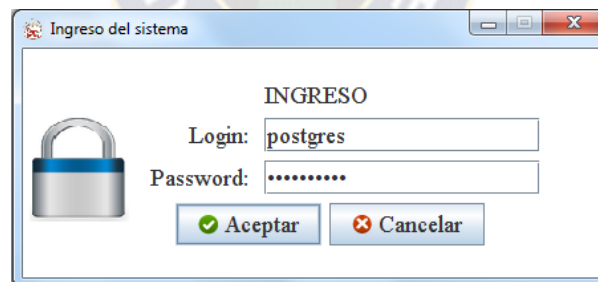


Figura 3.10. Ingreso al sistema.

Fuente: [Interface del sistema]

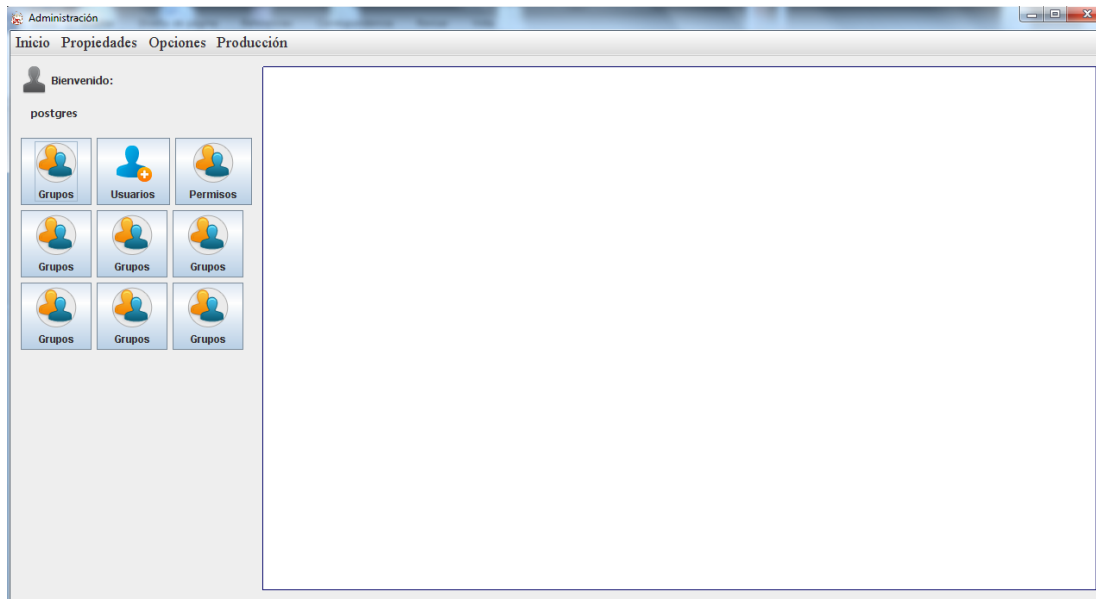


Figura 3.11. Sistema base.

Fuente: [Interface del sistema]

### 3.5. Planes de lanzamiento de acuerdo a los nuevos Sprints

Planificación de cada Sprint \* de acuerdo a la figura 2.6. del Capítulo II, propuesta de la gestión de la arquitectura de software en Scrum. Ver tabla 3.8.

Sprint	Fases	Agenda
Sprint 1	RF1	08-01-15 AL 10-01-15
Sprint 2	RF2	12-01-15 AL 24-01-15
Sprint 3	RF3, RF4, RF5, RF6	26-01-15 AL 14-02-15
Sprint 4	RF7	16-02-15 AL 21-02-15
Sprint 5	RF9, RF10, RF12, RF13, RF14, RF16, RF17, RF18, RF19	23-02-15 AL 14-03-15
Sprint 6	RF8, RF11, RF15	16-03-15 AL 02-05-15

Tabla 3.8. Planes de lanzamiento para los diferentes Sprints.

Fuente: [Elaboración propia]

### 3.6. Sprint 1

Desarrollamos el primer nivel de la pila de requerimientos:

- RF1: Diseñar la base de datos, que permita almacenar la elaboración de productos lácteos.

De acuerdo a la arquitectura diseñada procedemos a implementar la carga de la base de datos utilizando el diagrama relacional. Ver figura 3.3.

### 3.6.1. Sprint Backlog 1

	<b>Sprint</b>	<b>Inicio</b>	<b>Duración</b>
	1	08/01/2015	16 Hrs.

<b>Tarea</b>	<b>Estado</b>	<b>Responsable</b>	<b>Horas de trabajo</b>
Cargar Base de datos	Terminado	Junior Muñoz	2
Incluir secuencias	Terminado	Junior Muñoz	2
Incluir las funciones necesarias	Terminado	Junior Muñoz	4
Incluir vistas necesarias	Terminado	Junior Muñoz	4
Cargar datos de pruebas	Terminado	Junior Muñoz	4

Tabla 3.9. Primer sprint backlog realizado.

Fuente: [Elaboración propia]

### 3.7.Sprint 2

Desarrollamos el segundo nivel de la pila de requerimientos:

- RF2: Se debe manejar roles, usuarios, grupos de usuarios y permisos.

Se comienza detallando, las características principales de la siguiente manera:

Elaborar módulos los cuales puedan acceder a la base de datos del servidor Postgres mediante la autenticación de un usuario, a los esquemas desarrollados, a las tablas de los esquemas que se desarrollan, a los grupos, a los usuarios, y además de poder establecer permisos sobre tablas y esquemas.

Desarrollar el formulario de despliegue de las tablas elaboradas, dependiendo del esquema al que pertenece. Ver figura 3.12.

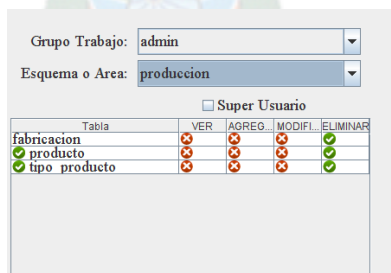


Figura 3.12. Ventana de despliegue de tablas elaboradas.

Fuente: [Interface del sistema]

Desarrollar el formulario de despliegue para la transacción de grupos. Ver figura 3.13.



Figura 3.13. Ventana de transacciones de grupos.

Fuente: [Interface del sistema]

Realizar la ventana para el manejo de permisos. Ver figura 3.14.

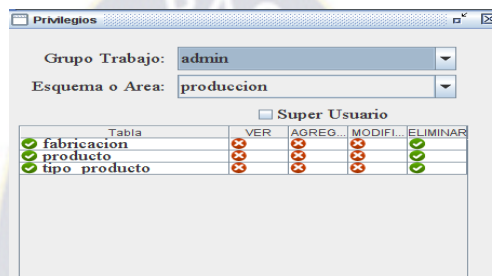


Figura 3.14. Ventana de manejo de permisos o privilegios.

Fuente: [Interface del sistema]

### 3.7.1. Prueba de Unidad

Se realizan las pruebas unitarias de los diferentes métodos presentados anteriormente, en las siguientes figuras se muestra ejemplos de cómo se realizaron las pruebas unitarias con la ayuda de JUnit. Ver figura 3.15.

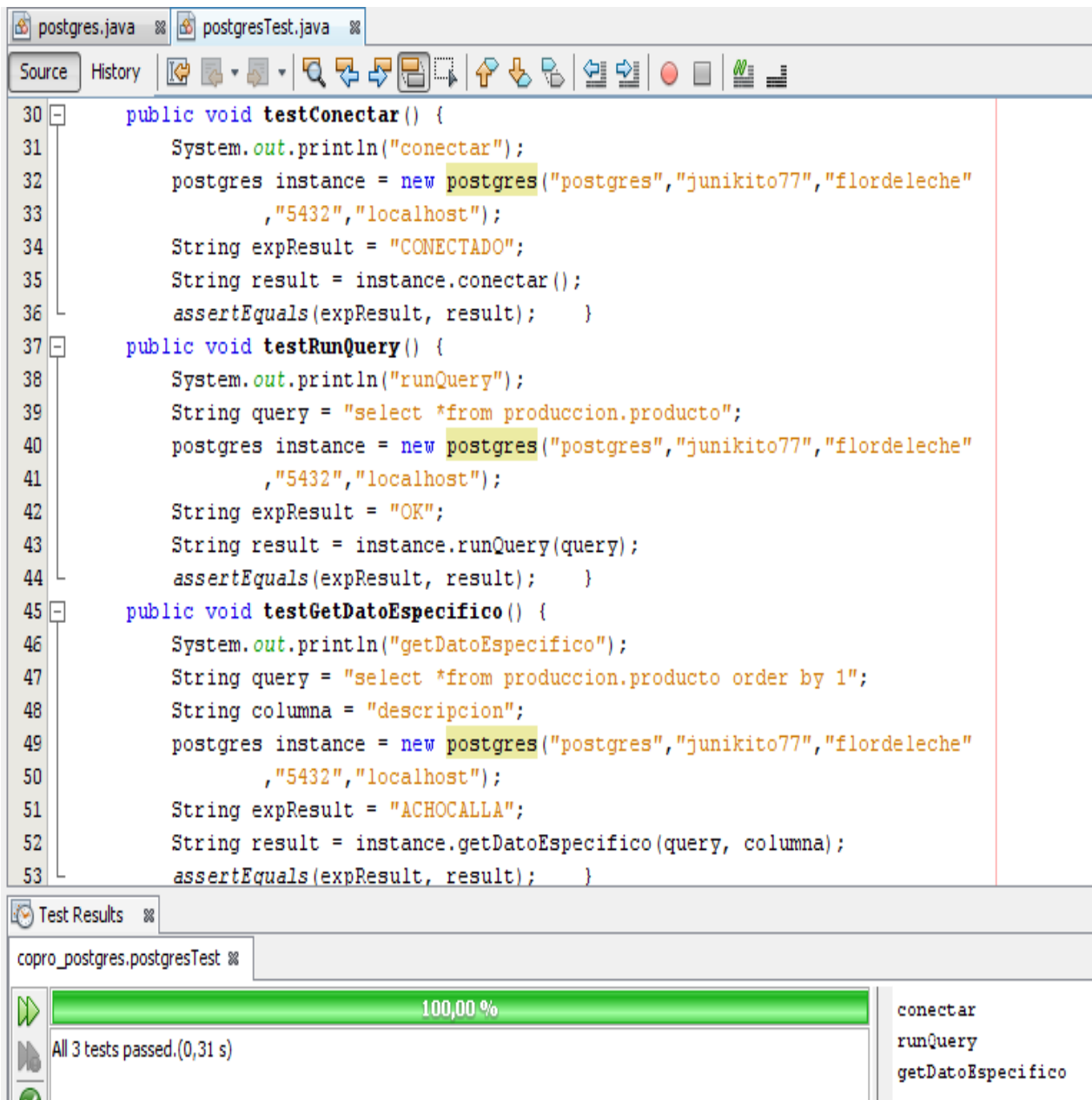


Figura 3.15. Pruebas del sprint 2

Fuente: [Pruebas del sistema]

### 3.7.2. Sprint Backlog 2

Sprint	Inicio	Duración		
2	12/01/2015	80 Hrs.		
Tarea	Estado	Responsable	Horas de trabajo	
Conexión a Postgres	Terminado	Junior Muñoz	16	

Despliegue de esquemas	Terminado	Junior Muñoz	16
Despliegue de tablas	Terminado	Junior Muñoz	16
Transacciones en grupos	Terminado	Junior Muñoz	16
Transacciones en usuarios	Terminado	Junior Muñoz	16
Manejo de permisos	Terminado	Junior Muñoz	10

Tabla 3.10. Segundo sprint backlog realizado.

Fuente: [Elaboración propia]

### 3.8.Sprint 3

Desarrollamos el tercer nivel de la pila de requerimientos:

- RF4: La inserción, modificación y eliminación de tipos de productos se debe tomar en cuenta la siguiente información (descripción, y la unidad) ver figura 3.16.

The screenshot shows a window titled 'Tipos de Producto'. At the top, it says 'Tipos de producto' and 'Codigo: #tipo\_producto#'. Below this are fields for 'Descripción:' and 'Unidad:' with a dropdown arrow. There are two buttons: 'Nuevo' (with a green checkmark) and 'Cancelar' (with a red X). Below the form is a table with columns: 'Codigo', 'Descripción', 'Unidad', 'Modificar', and 'Eliminar'. The table contains 11 rows of product types.

Codigo	Descripción	Unidad	Modificar	Eliminar
3	ACHOCALLAS	KILOS	●	●
7	CREMAS	LITRO	●	●
1	EDAM	KILOS	●	●
4	FLORECIDOS	UNIDAD	●	●
9	FRESCOS	KILOS	●	●
8	FRESCOS UN	UNIDAD	●	●
2	RACLETTE	KILOS	●	●
11	ROCA	KILOS	●	●
0	TILSIT	KILOS	●	●
6	YOGURT DIETA	FRASCO	●	●
5	YOGURT ENTERO	FRASCO	●	●

Figura 3.16. Ventana de transacciones de tipos de producto.

Fuente: [Interface del sistema]

- RF5: El registro de productos debe tomar en cuenta la siguiente información (descripción, relación-molde, y llevar una relación con los tipos de producto) (ver figura 3.17.)



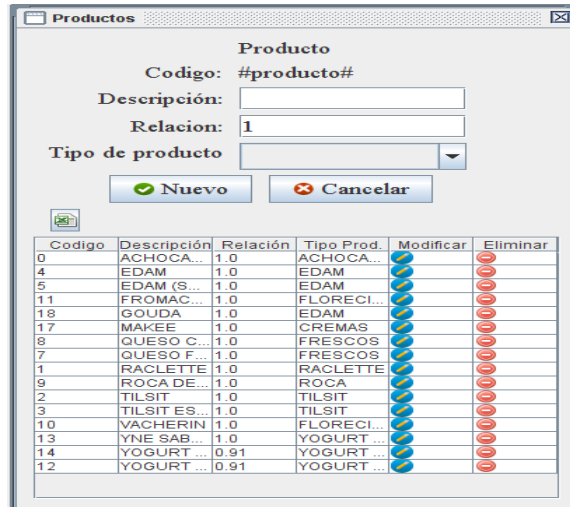


Figura 3.17. Ventana de transacciones de productos.

Fuente: [Interface del sistema]

- RF6: El sistema debe calcular de manera automática el rendimiento.
- RF3: La inserción, modificación, eliminación de la fabricación debe ser flexible y debe tomarse en cuenta (La fecha de producción, el proceso, la recepción de leche, el producto, los moldes, los kilos finales, las horas de pasteurización, las horas de fabricación, el responsable de fabricación, el pH, la altura en cm) ver figura 3.18.

Nº	Codigo	Descripción	Proceso	Fecha (d-m-A)	Rec. Leche	Kg	Nº moldes	pH	Acm	Responsable	Rendimie	Tiempo Past	Tiempo Fab.
69	9775	TILSIT	1	05-03-2015	630	29,56	36	0	25	SELIA MAMANI CHOQ	10,578	00:20:00	03:30:00
70	9776	TILSIT	1	11-03-2015	380	36,65	21	0	46	VILMA YURJA MAQUE	10,368	00:20:00	03:30:00
71	9777	TILSIT	1	16-03-2015	380	38,15	22	0	46	SELIA MAMANI CHOQ	9,961	00:20:00	03:30:00
72	9778	TILSIT	1	19-03-2015	380	37,95	22	0	46	VILMA YURJA MAQUE	10,013	00:20:00	03:30:00
73	9779	TILSIT	1	24-03-2015	630	58,88	36	0	25	NELY ESCOBAR CHOQ	10,700	00:20:00	03:30:00
74	9780	TILSIT	1	26-03-2015	380	38,77	22	0	46	SELIA MAMANI CHOQ	9,801	00:20:00	03:30:00
75	9781	TILSIT	1	30-03-2015	380	38,76	22	0	46	SELIA MAMANI CHOQ	9,804	00:20:00	03:30:00
76	9782	TILSIT	1	31-03-2015	630	62,82	36	0	25	NELY ESCOBAR CHOQ	10,029	00:20:00	03:30:00
77	9783	TILSIT ESPECIES(DIET)	1	09-03-2015	272	34,29	19	0	46	SELIA MAMANI CHOQ	7,932	00:15:00	00:15:00
78	9784	TILSIT ESPECIES(DIET)	1	23-03-2015	272	37,11	19	0	46	SELIA MAMANI CHOQ	7,330	00:15:00	00:15:00
79	9805	VACHERIN	1	03-03-2015	230	0	125	0	29	VILMA YURJA MAQUE	1,84	00:15:00	03:00:00
80	9806	VACHERIN	1	17-03-2015	230	0	126	0	29	VILMA YURJA MAQUE	1,825	00:15:00	03:00:00
81	9807	VACHERIN	1	24-03-2015	310	0	167	0	38	VILMA YURJA MAQUE	1,856	00:15:00	03:00:00
82	9808	VACHERIN	1	31-03-2015	230	0	134	0	29	VILMA YURJA MAQUE	1,716	00:15:00	03:00:00
83	9770	YNE SABORIZADO	1	05-03-2015	65	3	3	0	0	JUSTINA PEREZ HUAN	21,667	00:15:00	04:00:00
84	9771	YNE SABORIZADO	1	12-03-2015	105	5	5	0	0	JUSTINA PEREZ HUAN	21	00:15:00	04:00:00
85	9772	YNE SABORIZADO	1	19-03-2015	105	5	5	0	0	JUSTINA PEREZ HUAN	21	00:15:00	04:00:00
86	9773	YNE SABORIZADO	1	26-03-2015	105	5	5	0	0	JUSTINA PEREZ HUAN	21	00:15:00	04:00:00
87	9760	YOGURT NATURAL DI...	1	02-03-2015	74	73,71	81	0	0	JUSTINA PEREZ HUAN	1,004	00:30:00	05:00:00
88	9761	YOGURT NATURAL DI...	1	09-03-2015	93	90,09	99	0	0	JUSTINA PEREZ HUAN	1,032	00:30:00	05:00:00
89	9762	YOGURT NATURAL DI...	1	16-03-2015	93	93,73	103	0	0	JUSTINA PEREZ HUAN	0,992	00:30:00	05:00:00
90	9763	YOGURT NATURAL DI...	1	23-03-2015	93	91,91	101	0	0	JUSTINA PEREZ HUAN	1,012	00:30:00	05:00:00
91	9764	YOGURT NATURAL DI...	1	30-03-2015	99	96,46	106	0	0	JUSTINA PEREZ HUAN	1,026	00:30:00	05:00:00
92	9756	YOGURT NATURAL EN...	1	06-03-2015	100	98,28	108	0	0	JUSTINA PEREZ HUAN	1,018	00:15:00	04:00:00
93	9757	YOGURT NATURAL EN...	1	13-03-2015	141	141,05	155	0	0	JUSTINA PEREZ HUAN	1	00:15:00	04:00:00
94	9758	YOGURT NATURAL EN...	1	20-03-2015	148	154,7	170	0	0	JUSTINA PEREZ HUAN	0,957	00:15:00	04:00:00
95	9759	YOGURT NATURAL EN...	1	27-03-2015	138	142,87	157	0	0	JUSTINA PEREZ HUAN	0,966	00:15:00	04:00:00

Figura 3.18. Ventana de transacciones de fabricación

Fuente: [Interface del sistema]

### 3.8.1. Prueba de Unidad

Se realizan las pruebas unitarias de los diferentes métodos presentados anteriormente, en las siguientes figuras se muestra ejemplos de cómo se realizaron las pruebas unitarias con la ayuda de JUnit. Ver figura 3.19.

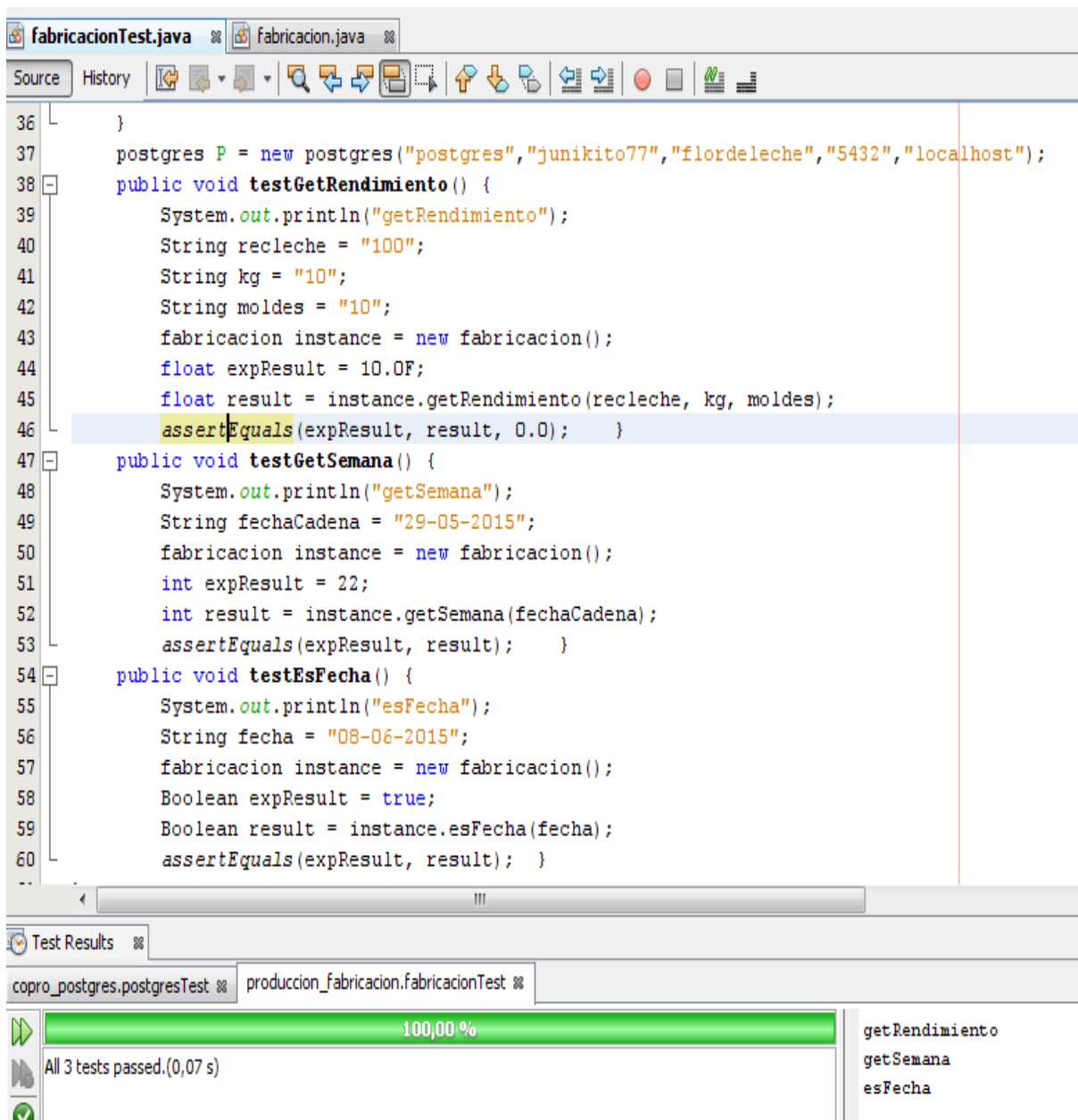


Figura 3.19. Pruebas del sprint 3

Fuente: [Pruebas del sistema]

### 3.8.2. Sprint Backlog 3

	<b>Sprint</b>	<b>Inicio</b>	<b>Duración</b>
	3	05/01/2015	120 Hrs.

<b>Tarea</b>	<b>Estado</b>	<b>Responsable</b>	<b>Horas de trabajo</b>
Registro de tipos de producto	Terminado	Junior Muñoz	15
Registro de productos	Terminado	Junior Muñoz	15
Registro de la fabricación	Terminado	Junior Muñoz	80
Algoritmo para el calculo del rendimiento	Terminado	Junior Muñoz	10

Tabla 3.11. Tercer sprint backlog realizado.

Fuente: [Elaboración propia]

### 3.9.Sprint 4

Desarrollamos el cuarto nivel de la pila de requerimientos:

- RF7: Toda la información debe de tener la posibilidad de ser exportada a una planilla Excel, ver figura 3.20.

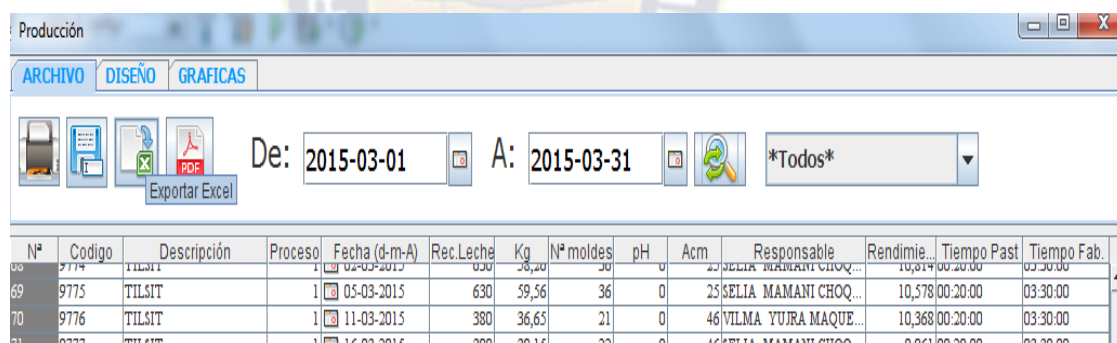


Figura 3.20. Opción de exportación a Excel.

Fuente: [Interface del sistema]

#### 3.9.1. Prueba de Unidad

Se realizan las pruebas unitarias de los diferentes métodos presentados anteriormente, en las siguientes figuras se muestra ejemplos de cómo se realizaron las pruebas unitarias con la ayuda de JUnit. Ver figura 3.21.

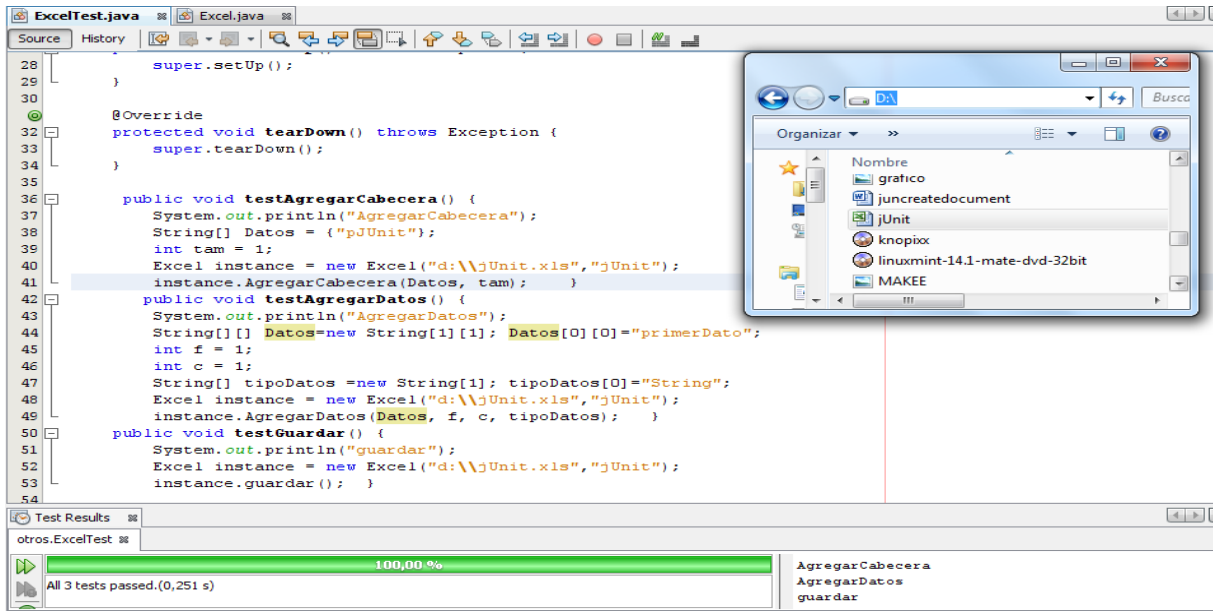


Figura 3.21. Pruebas del sprint 4

Fuente: [Pruebas del sistema]

### 3.9.2. Sprint Backlog 4

Sprint	Inicio	Duración		
4	16/02/2015	48 Hrs.		
Tarea	Estado	Responsable	Horas de trabajo	
Generar algoritmo general para exportación	Terminado	Junior Muñoz	30	
Exportación a Excel el formulario tipos de producto	Terminado	Junior Muñoz	6	
Exportación a Excel el formulario productos	Terminado	Junior Muñoz	6	
Exportación a Excel el formulario fabricación	Terminado	Junior Muñoz	6	

Tabla 3.12. Cuarto sprint backlog realizado

Fuente: [Elaboración propia]

### 3.10.Sprint 5

Desarrollamos el quinto nivel de la pila de requerimientos:

- RF9: La información para los reportes de fabricación debe tomarse en cuenta de acuerdo a las fechas de selección.

Se reutiliza el formulario de fabricación para así darle la opción de generar los reportes solicitados. Ver figura 3.22.

N°	Código	Descripción	Proceso	Fecha (d-m-A)	Rec. Leche	Kg	N° moldes	pH	Acm	Responsable	Rendimie.	Tiempo Past	Tiempo Fab.
1	9795	ACHOCALLA	1	03-03-2015	760	63,29	12	0	29,2	SELIA MAMANI CHOQUE	12,008	00:15:00	03:00:00
2	9796	ACHOCALLA	1	05-03-2015	380	34,93	6	0	46,1	NELY ESCOBAR CHOQUE	10,879	00:15:00	03:00:00
3	9797	ACHOCALLA	1	11-03-2015	760	68,69	12	0	29,1	SELIA MAMANI CHOQUE	11,064	00:15:00	03:00:00
4	9798	ACHOCALLA	1	27-03-2015	760	68,65	12	0	29	VELMA YUJRA MAQUERA	11,071	00:15:00	03:00:00
5	9785	EDAM	1	04-03-2015	380	38,8	23	0	46,5	VELIA MAMANI CHOQUE	9,794	00:15:00	03:10:00
6	9786	EDAM	1	06-03-2015	630	62,85	37	0	25	VELMA YUJRA MAQUERA	10,024	00:15:00	03:10:00
7	9787	EDAM	1	13-03-2015	380	36,64	23	0	46,1	SELIA MAMANI CHOQUE	10,371	00:15:00	03:10:00
8	9788	EDAM	1	20-03-2015	380	33,78	23	0	46	VELMA YUJRA MAQUERA	11,248	00:15:00	03:10:00
9	9789	EDAM	1	25-03-2015	630	61,74	37	0	25	SELIA MAMANI CHOQUE	10,204	00:15:00	03:10:00
10	9790	EDAM	1	27-03-2015	380	38,46	23	0	46	VELMA YUJRA MAQUERA	9,88	00:15:00	03:10:00
11	9811	EDAM (SUBSIDIO)	1	04-03-2015	1.440	136,22	26	0	56,3	VELMA YUJRA MAQUERA	10,571	01:20:00	03:15:00
12	9812	EDAM (SUBSIDIO)	1	16-03-2015	1.440	144,29	21	0	56	VELMA YUJRA MAQUERA	9,98	01:20:00	03:15:00
13	9813	EDAM (SUBSIDIO)	1	21-03-2015	760	80,07	11	0	29	NELY ESCOBAR CHOQUE	9,492	01:20:00	03:15:00
14	9815	EDAM (SUBSIDIO)	1	26-03-2015	1.440	149,41	21	0	56,2	BERGIO LEONARDO GABRIEL CHINO CO...	9,638	01:20:00	03:15:00
15	9814	EDAM (SUBSIDIO)	2	21-03-2015	380	39,34	2	0	46	SELIA MAMANI CHOQUE	9,659	01:20:00	03:15:00
16	9809	FROMACHELLI	1	10-03-2015	230	0	133	0	29	VELMA YUJRA MAQUERA	1,729	00:15:00	03:15:00
17	9810	FROMACHELLI	1	24-03-2015	310	0	167	0	38	VELMA YUJRA MAQUERA	1,856	00:15:00	03:15:00
18	9752	MAKEE	1	05-03-2015	80	20	0	0	0	JUSTINA PEREZ HUANCA	4,00	00:15:00	08:00:00
19	9753	MAKEE	1	12-03-2015	80	20	0	0	0	JUSTINA PEREZ HUANCA	4,00	00:15:00	08:00:00
20	9754	MAKEE	1	19-03-2015	60	15	0	0	0	JUSTINA PEREZ HUANCA	4,00	00:15:00	08:00:00
21	9755	MAKEE	1	27-03-2015	100	25	0	0	0	JUSTINA PEREZ HUANCA	4,00	00:15:00	08:00:00
22	9765	QUESO CRIOLLO	1	03-03-2015	300	0	96	0	0	JUSTINA PEREZ HUANCA	3,125	00:30:00	04:00:00
23	9766	QUESO CRIOLLO	1	10-03-2015	200	0	67	0	0	JUSTINA PEREZ HUANCA	2,985	00:30:00	04:00:00
24	9767	QUESO CRIOLLO	1	17-03-2015	300	0	95	0	0	JUSTINA PEREZ HUANCA	3,158	00:30:00	04:00:00
25	9768	QUESO CRIOLLO	1	24-03-2015	250	0	87	0	0	JUSTINA PEREZ HUANCA	2,874	00:30:00	04:00:00
26	9769	QUESO CRIOLLO	1	31-03-2015	300	0	100	0	0	JUSTINA PEREZ HUANCA	3,00	00:30:00	04:00:00
27	9816	QUESO FRESCO(PRENSADO)	1	02-03-2015	1.430	171,43	24	0	56,4	NELY ESCOBAR CHOQUE	8,342	01:20:00	03:00:00
28	9817	QUESO FRESCO(PRENSADO)	1	05-03-2015	1.430	165,56	24	0	56,1	NELY ESCOBAR CHOQUE	8,632	01:20:00	03:00:00

Figura 3.22. Reporte generado por fecha de selección

Fuente: [Interface del sistema]

### 3.10.1. Sprint Backlog 5

Sprint	Inicio	Duración
5	23/02/2015	120 Hrs.

Tarea	Estado	Responsable	Horas de trabajo
Generar consultas para cada reporte	Terminado	Junior Muñoz	80
Acomodar los formularios para respectivo reporte	Terminado	Junior Muñoz	20
Generar los reportes	Terminado	Junior Muñoz	20

Tabla 3.13. Quinto sprint backlog realizado

Fuente: [Elaboración propia]

### 3.11.Sprint 6

Desarrollamos el sexto nivel de la pila de requerimientos:

- RF11: La graficas de rendimientos deben ser presentadas en dos formas: presentación de la gráfica y la presentación en el reporte, ver figura 3.23.

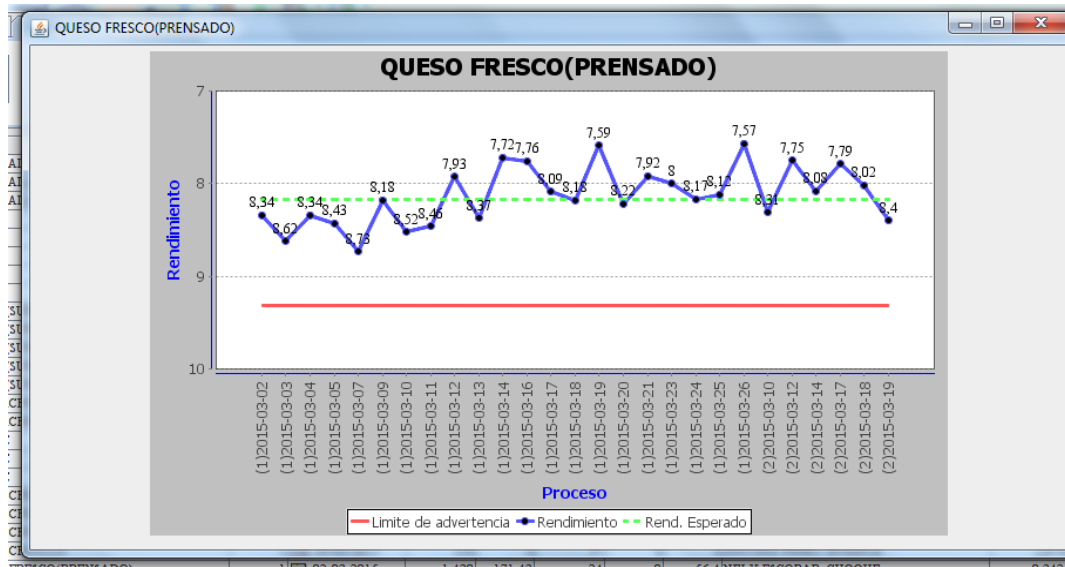


Figura 3.23. Gráficas de rendimientos.

Fuente: [Interface del sistema]

- RF8: Los reportes para poder ser editados se debe exportar a un formato WORD, ver figura 3.24.

N°	Código	Descripción	Proceso	Fecha (d-m-A)	Rec. Leche	Kg	Nº moldes	pH	Acm	Responsable
1	9795	ACHOCALLA	1	03-03-2015	760	63,29	12	0	29,2	SELIA MAMANI CHOQUE
2	9796	ACHOCALLA	1	07-03-2015	380	34,93	6	0	46,1	NELY ESCOBAR CHOQUE
3	9797	ACHOCALLA	1	11-03-2015	760	68,69	12	0	29,1	SELIA MAMANI CHOQUE
4	9785	EDAM	1	04-03-2015	380	38,8	23	0	46	SELIA MAMANI CHOQUE
5	9786	EDAM	1	06-03-2015	630	62,85	37	0	25	VILMA YUJRA MAQUERA

Figura 3.24. Reporte en formato Word

Fuente: [Interface del sistema]

- RF15: La información de los niveles de pH se debe poder presentar en una gráfica y en los reportes, ver figura 3.25.



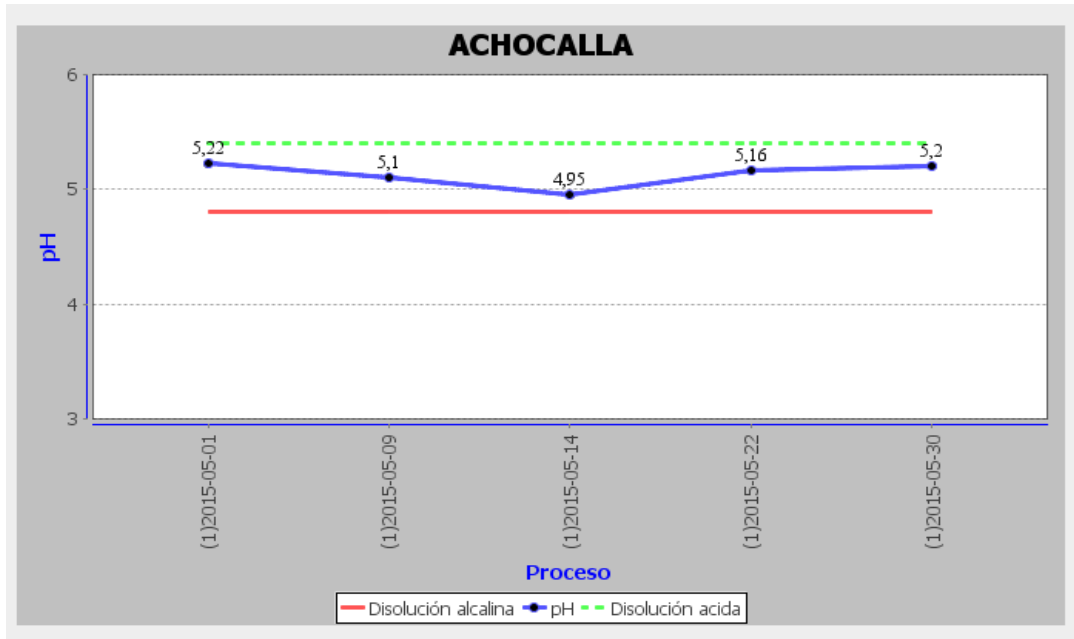


Figura 3.25. Gráfica de pH.

Fuente: [Interface del sistema]

### 3.11.1. Sprint Backlog 6

Sprint	Inicio	Duración
6	16/03/2015	280 Hrs.

Tarea	Estado	Responsable	Horas de trabajo
Gráficas de rendimientos	Terminado	Junior Muñoz	120
Gráficas de pH			120
Reporte WORD de rendimientos	Terminado	Junior Muñoz	40

Tabla 3.14. Sexto sprint backlog realizado.

Fuente: [Elaboración propia]

### 3.11.2. Diagrama navegacional

El diagrama navegacional, permite identificar de qué forma se adecuaron los formularios desarrollados, y de esta manera el usuario puede identificar, que elementos se encuentran en el software y donde, además de esta manera se termina el desarrollo con la metodología Scrum. Ver figura 3.26.



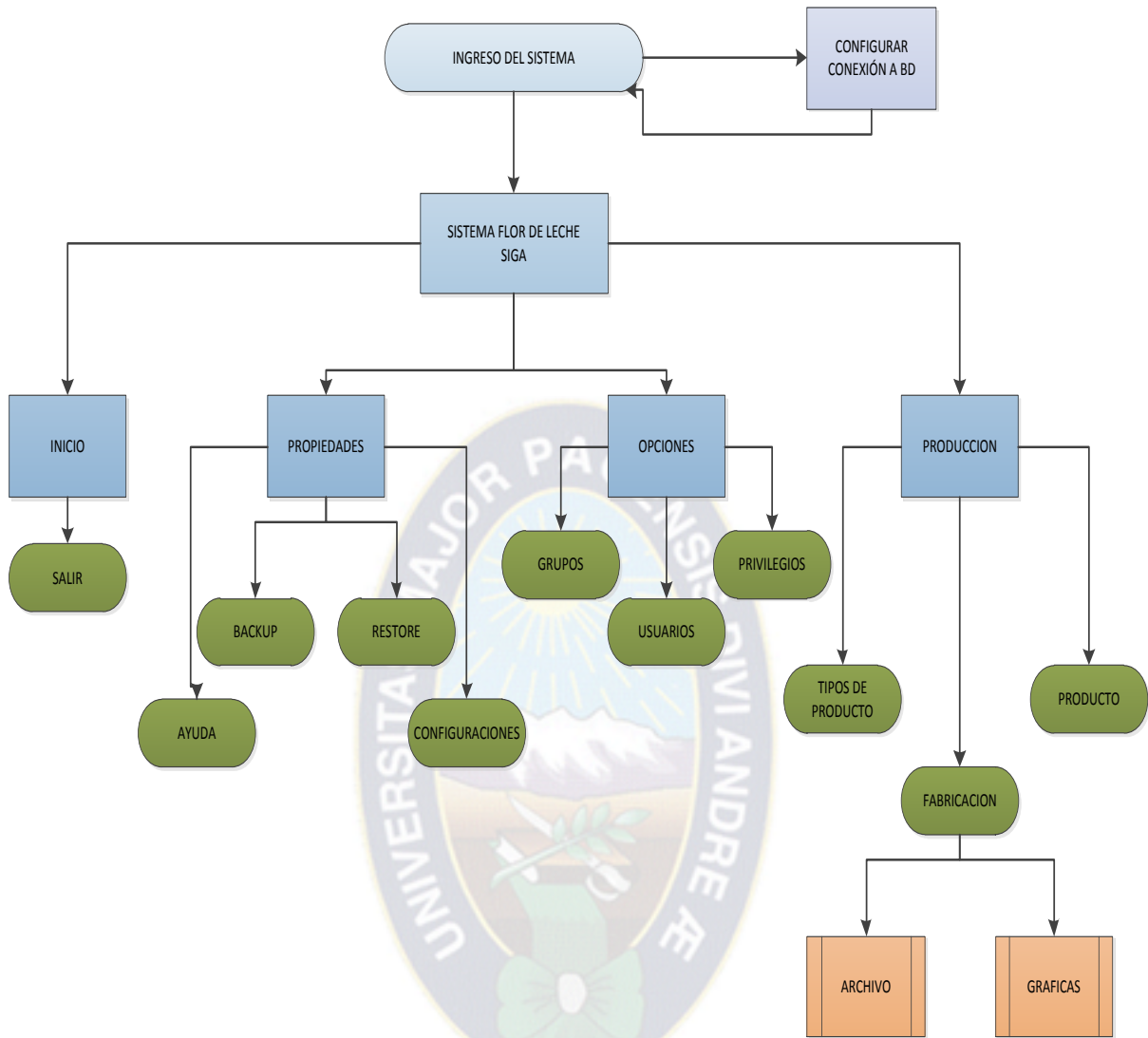


Figura 3.26. Diagrama navegacional.

Fuente: [Elaboración propia]

**CAPITULO IV**  
**PRUEBAS Y EVALUACIÓN DEL SISTEMA**

**4.1.Introducción**

Las métricas de calidad aporta una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. A veces, cuando se intentan obtener medidas precisas de la calidad del software se acaba frustrado por la naturaleza subjetiva de esta actividad. Para resolver este problema, se buscan medidas cuantitativas de la calidad del software, para poder llevar a cabo un análisis objetivo.

Pero de alguna forma es un poco complicado medir la calidad del software de forma exacta, ya que cada medida es parcialmente imperfecta. Las medidas de calidad siempre son indirectas, ya que no se mide directamente la calidad sino algunas de sus manifestaciones. El factor que lo complica es la relación precisa entre la variable que es medida y la calidad del software.

Para determinar ciertas medidas que enmarquen el sistema sobre un conjunto de valores aceptables de calidad, nos remitimos a medir la calidad del sistema de manera directa, tomando en cuenta mediciones sobre el proceso mismo del desarrollo, es decir sobre el modelo de análisis, el modelo de diseño, utilizando las métricas técnicas.

**4.2.Calidad de software**

Realizamos el cálculo del punto función de acuerdo a los parámetros de medición, (ver tabla4.1).

Parámetros de medición	Cuenta	Factor de ponderación						
			Simple	Medio	Complejo			
Número de entradas de usuario	9	x	5	4	3	=	36	
Número de salidas de usuario	13	x	6	9	3	=	117	
Número de peticiones de usuario	2	x	1	13	0	=	26	
Número de archivos	4	x	1	9	5	=	36	
Número de interfaces externas	1	x	6	4	0	=	4	
Cuenta total	→						219	

Tabla 4.1. Cálculo de cuenta total de la métrica de punto función.

Fuente:[Elaboración propia]

Nro.	Preguntas	Factor de ajuste
1	El sistema requiere respaldo y recuperación confiables	3
2	Se requieren comunicaciones de datos especializadas para transferir información a la aplicación, u obtener de ella	2
3	Hay funciones distribuidas de procesamiento	4
4	El desempeño es critico	1
5	El sistema se ejecutara en un entorno existente que tiene uso pesado de operaciones	2
6	El sistema requiere entrada de datos en línea	2
7	La entrada de datos en línea requiere de la transacción de entrada se construya en varias pantallas u operaciones	2
8	Los archivos lógicos internos se actualizan en línea.	1
9	Las entradas, las salidas, los archivos o las consultas son complejos.	2
10	Es complejo el procesamiento interno.	3
11	El código diseñado será reutilizable	5
12	Se incluyen la conversión e instalación en el diseño	4
13	Está diseñado el sistema para instalaciones múltiples en diferentes organizaciones	5
14	La aplicación está diseñada para aplicar el cambio y para que el usuario lo use fácilmente	5
<b>Total</b>		<b>41</b>

Tabla 4.2. Valor de ajuste de complejidad de punto de función.

Fuente:[Elaborado de acuerdo a PRESS]

**Cuenta total:** Es el total de puntos de función sin ajustar.

**0.01:** es el error mínimo aceptable de la complejidad.

$\sum F_i$ : Es la sumatoria de los factores de complejidad del proceso que pueda variar en el rango de:  $0 \leq \sum F_i \leq 70$ ,  $i=1, \dots, 14$

Si los factores de complejidad de proceso no influyen en nada  $\sum F_i=0$  y si influyen mucho  $\sum F_i=70$

**0.65:** es el pie de corrección.

El valor 219 es la sumatoria de los resultados parciales que son los productos de las cuentas por el peso asignado (Simple, medio y complejo), y para realizar el análisis utilizamos el peso medio.

Luego reemplazamos los datos en la relación de Punto de Función y se obtiene el siguiente resultado:

$$PF=219*(0.65+(0.01*41))=231.14$$

Haciendo un análisis con la escala de Punto de Función se concluye que sistema tiene la funcionalidad Buena.

Escala	Observación
PF>300	Optima
200<PF<300	Buena
100<PF<200	Suficiente
PF<100	Deficiente

Tabla 4.3. Escala de Punto de Función

Fuente: [Elaboración propia]

Calculando el ajuste tenemos:

El valor promedio de Fi(promedio calculado)=41

Valor máximo de Fi(máximo)=70

La cuenta total del facto de ponderaciones = 219

Reemplazando estos valores en la ecuación:

$$PF(\text{máximo})=219*(0.65*(0.01*70))=295.65$$

Sacando el promedio de estos dos resultados tenemos:

$$\text{Prom}=\frac{231.14}{295.65} = 0.7852$$

Por lo tanto se tiene un 78.52% de funcionalidad

#### 4.2.1. Confiabilidad

Las fallas o errores presentados por el sistema no se presentan frecuentemente, en la fase inicial del sistema se realizó la corrección y el ajuste necesario para corregir los errores que presento el sistema.

Para determinar la confiabilidad del sistema se especifica el instante en el que comienza a funcionar determinado por  $t_0=0$ . A partir de este momento se observa el trabajo del sistema hasta que se introduzca una falla en el instante T que se va aproximando a una variable aleatoria continua, que nos determina la confiabilidad en términos probabilísticos. Entonces se tiene las siguientes probabilidades:

$$P(T \leq t) = F(t) \quad \dots(1) \text{Probabilidad de fallas}$$

$$P(T > t) = 1 - F(t) \dots(2) \text{Probabilidad de trabajo sin fallas}$$

Se calcula estas probabilidades mediante la distribución exponencial. Donde, punto función, resultado obtenido en el cálculo realizado en un punto anterior a este.

Fueron realizadas 8 ejecuciones por mes en un periodo de  $t=14$ .

Con un margen de error  $\lambda=1/10$ . Entonces se tiene.

$F(t) = \text{Puntofunción} * e^{(-\lambda*t)}$  ...ecuación de confiabilidad

$$F(t) = 0.7852 * e^{(-\frac{1}{10}*14)} = 0.19$$

$$\mathbf{F(t)=0.19}$$

Por tanto, el sistema presenta una probabilidad de fallo de un 19%. En lo posible el sistema es capaz de recuperar la información.

#### 4.2.2. Mantenimiento

Se calcula el índice de madurez del software (IMS), estableciendo los cambios que ocurrieron con la versión del producto.

MT: Nro. De módulos en la versión actual.8

Fc: Nro. De módulos en la versión actual que se han cambiado.2

Fa: Nro. De módulos en la versión actual que se han añadido.1

Fa: Nro. De módulos en la versión actual que se han eliminado.0

El IMS está dada por:

$$\text{IMS} = (\text{MT} - (\text{Fc} + \text{Fa} + \text{Fe})) / \text{MT}$$

$$\text{IMS} = (8 - (2 + 1 + 0)) / 6 = 0.833$$

$$\mathbf{\text{IMS}=0.833}$$

Por tanto, el índice de Madurez del software es de un 83.3%, estableciendo los resultados obtenidos tienen una estabilidad alta de la evolución en la versión lograda

#### 4.3. Estimación de costos con COCOMO

De acuerdo a las ecuaciones presentadas en el capítulo II, acerca de la estimación de costos, se propone utilizar la herramienta propia de la estimación de costos con COCOMO, la cual utilizando la cantidad de líneas programadas, podremos estimar correctamente el costo del software.

La herramienta es parte de USC Viterbi School Engineering, la herramienta permite calcular la estimación del costo, de acuerdo las líneas de programación, además de señalar la complejidad

del proyecto, en fin se la figura 4.1., se presenta la adición de datos que son necesarios para la estimación.

**Software Size**    Sizing Method: Source Lines of Code

**SLOC**    % Design Modified    % Code Modified    % Integration Required    Assessment and Assimilation (0% - 8%)    Software Understanding (0% - 50%)    Unfamiliarity (0-1)

New: 1051  
 Reused: 327    0    0    85    5  
 Modified: 380    15    25    90    7    45    1

**Software Scale Drivers**

Precedentedness: Nominal    Architecture / Risk Resolution: Nominal    Process Maturity: Nominal  
 Development Flexibility: Nominal    Team Cohesion: Nominal

**Software Cost Drivers**

**Product**  
 Required Software Reliability: Nominal  
 Data Base Size: Nominal  
 Product Complexity: Nominal  
 Developed for Reusability: High  
 Documentation Match to Lifecycle Needs: Nominal

**Personnel**  
 Analyst Capability: Nominal  
 Programmer Capability: Nominal  
 Personnel Continuity: Nominal  
 Application Experience: Nominal  
 Platform Experience: High  
 Language and Toolset Experience: Nominal

**Platform**  
 Time Constraint: Nominal  
 Storage Constraint: Nominal  
 Platform Volatility: Low

**Project**  
 Use of Software Tools: Nominal  
 Multisite Development: Nominal  
 Required Development Schedule: Nominal

Figura 4.1. Inserción de datos para COCOMO

Fuente [Herramienta COCOMO]

La figura 4.2., muestra los resultados obtenidos realizando el cálculo del formulario.

Maintenance: Off

**Software Labor Rates**  
 Cost per Person-Month (Dollars): 450  
 Calculate

---

**Results**

**Software Development (Elaboration and Construction)**    **Staffing Profile**

Effort = 3.8 Person-months    Your project is too small to display a staffing profile due to truncation.  
 Schedule = 5.7 Months  
 Cost = \$1711

Total Equivalent Size = 1469 SLOC

**Acquisition Phase Distribution**

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.2	0.7	0.3	\$103
Elaboration	0.9	2.1	0.4	\$411
Construction	2.9	3.6	0.8	\$1301
Transition	0.5	0.7	0.6	\$205

**Software Activity Distribution (Person-Months)**

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.3	0.1
Environment/CM	0.0	0.1	0.1	0.0
Requirements	0.1	0.2	0.2	0.0
Design	0.0	0.3	0.5	0.0
Implementation	0.0	0.1	1.0	0.1
Assessment	0.0	0.1	0.7	0.1
Deployment	0.0	0.0	0.1	0.1

Your output file is [http://csse.usc.edu/tools/data/COCOMO\\_June\\_11\\_2015\\_22\\_15\\_02\\_229743.txt](http://csse.usc.edu/tools/data/COCOMO_June_11_2015_22_15_02_229743.txt)

Figura 4.2. Resultado de estimación de costos

Fuente: [Herramienta COCOMO]

Como podemos observar el esfuerzo es de 3.8 personas-mes, el tiempo de desarrollo 5.7 meses y el costo final 1711\$, se presenta la tabla 4.4., donde se detalla los costos y el personal:



<b>Fase</b>	<b>Esfuerzo(persona- mes)</b>	<b>Tiempo (mes)</b>	<b>Promedio de personal</b>	<b>Costo en Dolares</b>
Inicio	0,2	0,7	0,3	\$103
Elaboración	0,9	2,1	0,4	\$411
Construcción	2,9	3,6	0,8	\$1301
Transición	0,5	0,7	0,6	\$205
<b>Total</b>	<b>3,8</b>	<b>5,7</b>		<b>\$1711</b>

Tabla 4.4. Detalle de costos.

Fuente: [Herramienta COCOMO]

Así de esta manera, finalizamos la estimación de costos, teniendo un costo final de \$1711,00 con un total de 4 personas como desarrolladores.

#### **4.4. Seguridad del sistema**

La seguridad que se maneja principalmente es a niveles de software, se tiene como inicio la creación de usuarios de sistema, los cuales te permiten iniciar sesión en el sistema operativo Windows 7. Seguidamente la seguridad de la base de datos Postgres, donde se crean usuarios a nivel de sistema operativo y permitiendo a estos usuarios pertenecer a grupos, los cuales tienen privilegios a los esquemas y finalmente a las tablas.

##### **4.4.1. Controles de acceso**

Debido a que se utiliza un sistema operativo Windows 7, control de acceso en el sistema es mediante usuarios de sistema, los cuales son separados y permiten separar su información de una sesión a otra, en este caso se cuenta con un control de acceso denominado Gerencia de producción (GPR).

##### **4.4.2. Identificación y autenticación**

La identificación y autenticación es la primera línea de código, el cual cada usuario tiene un login y un password en el sistema, los usuarios pueden autenticarse y mediante la consulta a la Base de datos el usuario podrá o no acceder al sistema, es así que también existen varios niveles de usuario, como por ej. El usuario Postgres, que es creado automáticamente al momento de instalar el sistema, tiene todos los privilegios de realizar cualquier transacción que se requiera.



#### 4.4.3. Roles

Debido a que la cantidad de usuarios se puede agrandar con el tiempo, se plantea la idea de manejar por grupos a todos los usuarios, en este caso se plantea inicialmente dos grupos los cuales tienen diferentes roles:

Producción → Teniendo los permisos del manejo total de toda el área de producción.

Admin → El rol Admin lleva consigo un súper-usuario debido a que es el administrador del sistema tiene todos los privilegios del sistema.



## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Introducción

A la conclusión del presente proyecto se detallan las conclusiones y recomendaciones. Para llevar a cabo este trabajo, se recabo información exhaustiva del proceso de desarrollo de software, áreas de conocimiento de ingeniería de software de sistemas y otras disciplinas dentro de la informática, se leyó bibliografía de libros, proyectos de grado similares y páginas web relacionadas con el tema.

#### 5.2. Conclusiones

Se ha desarrollado e implementado una herramienta software para el control de la elaboración de productos lácteos en la empresa Flor de Leche S.R.L., cumpliendo con los siguientes objetivos específicos:

- Implementar tecnologías de arquitectura de software para desarrollar un sistema escalable.

Se desarrolló un arquitectura de software que permite a cualquier otro aplicativo adecuarse fácilmente al sistema, así de esta manera utilizando metodologías de diseño y desarrollo como SCRUM, UML y ATAM.

- Obtener informes confiables, donde se puedan hacer comparaciones con datos pasados, de esta manera pronosticar los rendimientos, y tomar buenas decisiones.

Se obtienen reportes confiables, donde el usuario fácilmente puede comparar datos pasados, controlar la elaboración de los productos, controlar los rendimientos, y el control de los niveles de pH.

- Unir toda la información de todas las planillas para lograr integrar los datos y de esta manera no perder información.

Se unifico toda la información de gerencia de producción, ahora todos los datos relacionados y evitando siempre la redundancia, toda la información se guarda en la base de datos, teniendo así de esta manera mucha confianza sobre el resguardo de los datos.

- Agilizar el llenado de datos, para lograr mejores tiempos en el trabajo de gerencia de producción.

Se diseñó una planilla de ingreso de información, donde fácilmente es muy amigable el sistema al momento de adicionar y guardar los datos.

- Integrar la información en un solo motor de base de datos confiable.

Toda la información se integró en el motor de base de datos Postgres, la confiabilidad y el prestigio que tiene este motor de base de datos es muy alta, por lo tanto se confía plenamente la información que resguarda.

- Diseño e implementación de una Base de Datos.

El diseño e implementación se lo realizo de una forma que se puedan adherir fácilmente otros subsistemas.

De esta manera se cumple con el objetivo general:

“Desarrollar un sistema para realizar el control eficiente de la producción y pH de los productos lácteos procesados”.

### **5.3. Recomendaciones**

El software está desarrollado y orientado al uso de personas que tengan conocimiento del área de Administración y Gerencia de Producción, esto con el fin de mantener la integridad de la información.

Si bien el sistema cuenta con un nivel de confiabilidad y seguridad, es necesario realizar acciones para permitir mantener la madurez del sistema, para ello se recomienda:

Realizar un mantenimiento preventivo periódicamente, esto con el fin de permitir mejoras funcionales del sistema.

El módulo de administración de usuarios, asigna una clave maestra por defecto, que solo el administrador conoce, es recomendable que al finalizar la instalación se realice la asignación de los perfiles de los grupos correspondientes para los módulos.

Se recomienda realizar la continuidad del proyecto, agregando nuevos módulos, como se van mencionando, SEPRO (Servicios a la producción), afinamiento (Control de productos en sótanos), Inventarios de productos e insumos, Inventarios en embalaje, ventas y contabilidad, complementando con estos módulos, se obtendría un sistema totalmente robusto.

Por motivos de seguridad y desconfianza de la institución no se consideró el uso de páginas web, ya que la información que se maneja es muy valiosa para ellos y no hubo la necesidad de plantearla.

## Referencias Bibliográficas

### Referencia bibliográfica

- [ACM01] ARIÑEZ, M.(2001), Sistema de Información Integrado PIL-Chuquisaca S.A. (SII-PIL Chuquisaca),
- [ANM06] APAZA, M.(2006), Sistema de Información para la Producción Planificada DELIZIA, UMSA Biblioteca de Informática, Consultado Octubre 07, 2014.
- [FDL16] Plan de desarrollo organizacional 2014-2016, Flor de leche Srl.
- [FFDLQY11]Formula elaborado por GPR, Flor de leche Srl.
- [FRQY95] Fórmula matemática para el Cálculo del rendimiento, Flor de Leche S.R.L., consultado Julio 5, 2014
- [FXS08]Flexibilidad con Scrum, Principios de diseño e implantación de campos de Scrum, Autor: Juan Palacio.Edición Octubre 2008
- [MBA97] MACHICADO, A.(1997), Sistema de Almacenes y Producción SOCOVIAL, UMSA Biblioteca de Informática, Consultado Octubre 07, 2014.
- [MMPR12]CHAVEZ, E,(2013) tesis: modelo matemático para predecir el rendimiento del queso a partir de la composición química de la dieta, Consultado Octubre 04,2014
- [PURR10]RODRIGUEZ, R.(2010) Pruebas Unitarias,Universidad Nacional de San Juna (Consultado Marzo,2015)
- [PRESS] Pressman Roger, 5ta Ingeniería de software

### Referencia WEB

- [APO09] Árbol de problemas y Objetivos, Gobierno de Chile, Consejo Nacional de la Cultura y las Artes. Guía para la gestión de proyectos culturales. Valparaíso, 2009, extraído el 06 de Octubre de 2014.
- [http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Arbol\\_problemas\\_objetivo\\_s.doc](http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Arbol_problemas_objetivo_s.doc)
- [CAFU14]Calidad de Software, funcionalidad y usabilidad, (Consultado Abril, 2015)
- <http://albertolacalle.com/hci/funcionalidad-usabilidad.htm>
- [CBML] Curso breve de marco lógico (visión general del curso), extraído el 08 de Octubre de 2014

[http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.solucionesong.org%2Fimg%2Fforos%2F4ce164c4dd0de%2FMarco\\_Logico.ppt&ei=zfs1VJv3PPPksATfsYK4BQ&usg=AFQjCNHXLx0MJJeOWJqSSe9pnDYLNbNw-IA&sig2=2U\\_NiNQyVITugG1ToMHwrA&bvm=bv.76943099,d.cWc&cad=rja](http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.solucionesong.org%2Fimg%2Fforos%2F4ce164c4dd0de%2FMarco_Logico.ppt&ei=zfs1VJv3PPPksATfsYK4BQ&usg=AFQjCNHXLx0MJJeOWJqSSe9pnDYLNbNw-IA&sig2=2U_NiNQyVITugG1ToMHwrA&bvm=bv.76943099,d.cWc&cad=rja)

[CDA15] Andrea Delgado, Alberto Castro, Martín Germán, Universidad de la República, Facultad de Ingeniería, Instituto de Computación Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method), extraído 12 de Enero de 2015, {adelgado, acastro, mgerman}@fing.edu.uy

[CDS12] Calidad de Software, (Consultado Abril, 2015)  
[http://es.wikipedia.org/wiki/Calidad\\_de\\_software](http://es.wikipedia.org/wiki/Calidad_de_software)

[CLD10] Cliente Servidor de dos capas, (Consultado Enero, 2015)  
<http://www.monografias.com/trabajos89/cliente-servidor-dos-capas/cliente-servidor-dos-capas.shtml>

[COSC95], COCOMO, University of Southern California. (Consultado Marzo, 2015)  
[http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)

[ELQU04] Elaboración de Quesos (2004), extraído el 06 de Octubre de 2014  
<http://www.poncelet.es/enciclopedia-del-queso/elaboracion.html>

[FDL13] Empresa Flor de Leche S.R.L. (2013), extraído el 07 de Octubre de 2014  
<http://www.flordeleche.com.bo>

[IMPH00] Importancia del pH en las industrias y módulo de laboratorio (2000), extraído el 05 de Octubre de 2014  
[http://www.biblioteca.udep.edu.pe/bibvirUDEP/tesis/pdf/1\\_197\\_184\\_140\\_1851.pdf](http://www.biblioteca.udep.edu.pe/bibvirUDEP/tesis/pdf/1_197_184_140_1851.pdf)

[JLCH14] Jarra de Leche "Calf & Half" (2014), extraído el 03 de Noviembre de 2014  
<http://www.curiosite.es/scripts/product/esproduct.php?idproducto=48628033>

[JVE14] Java About, (Consultado Enero, 2015)  
[https://www.java.com/en/download/faq/whatis\\_java.xml](https://www.java.com/en/download/faq/whatis_java.xml)

[MAML12] Matriz del marco lógico (2012), extraído el 07 de Octubre de 2014  
[http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Matriz\\_de\\_Marco\\_Logico.pdf](http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Matriz_de_Marco_Logico.pdf)

[MSO15] Modelado de sistemas con UML, extraído el 12 de Enero 2015.

<http://www.monografias.com/Modelado de Sistemas con UML>

[PCB15] Pruebas de caja Blanca, (Consultado Abril, 2015)

[http://es.wikipedia.org/wiki/Pruebas\\_de\\_caja\\_blanca](http://es.wikipedia.org/wiki/Pruebas_de_caja_blanca)

[PGS12] PostgreSQL, (Consultado Enero, 2015)

<http://www.postgresql.org/about/>

[PHCA08] El pH en la conservación de alimentos

<https://es.scribd.com/doc/19649453/El-pH-en-la-conservacion-de-alimentos>

[PJ11] John Pruitt, Personal Scrum, Perspectives on software development 2011, extraído el 01 de Octubre de 2014. <http://blog.jgpruitt.com/2011/04/10/personal-scrum/>

[QPYC12] Queso partido y cortado (2012), extraído el 03 de Noviembre de 2014

<http://aquienseretaro.com/2012/12/14/el-queso-mas-viejo-del-mundo/>

[SSA12] R. Sullivan. "Scrum and Architecture", extraído el 06 de Octubre de 2014 .

<http://bit.ly/sg30r7>

[SBD14] Análisis Seguridad de bases de datos PostgreSQL, Septiembre, 2014, (Consultado Abril, 2015)

<http://www.byspel.com/seguridad-de-bases-de-datos-postgresql/>

[SSI12] Seguridad en un sistema de Información, (Consultado Abril, 2015)

<http://www.monografias.com/trabajos/seguinfo/seguinfo.shtml>



The background features a large, faded watermark of the University of the Pacific logo. The logo is an oval shape containing a sun with rays, a mountain range, and a green banner with a white cross. The text "UNIVERSITAS MAJOR PACENSIS DIVI ANDREAE" is written around the perimeter of the oval.

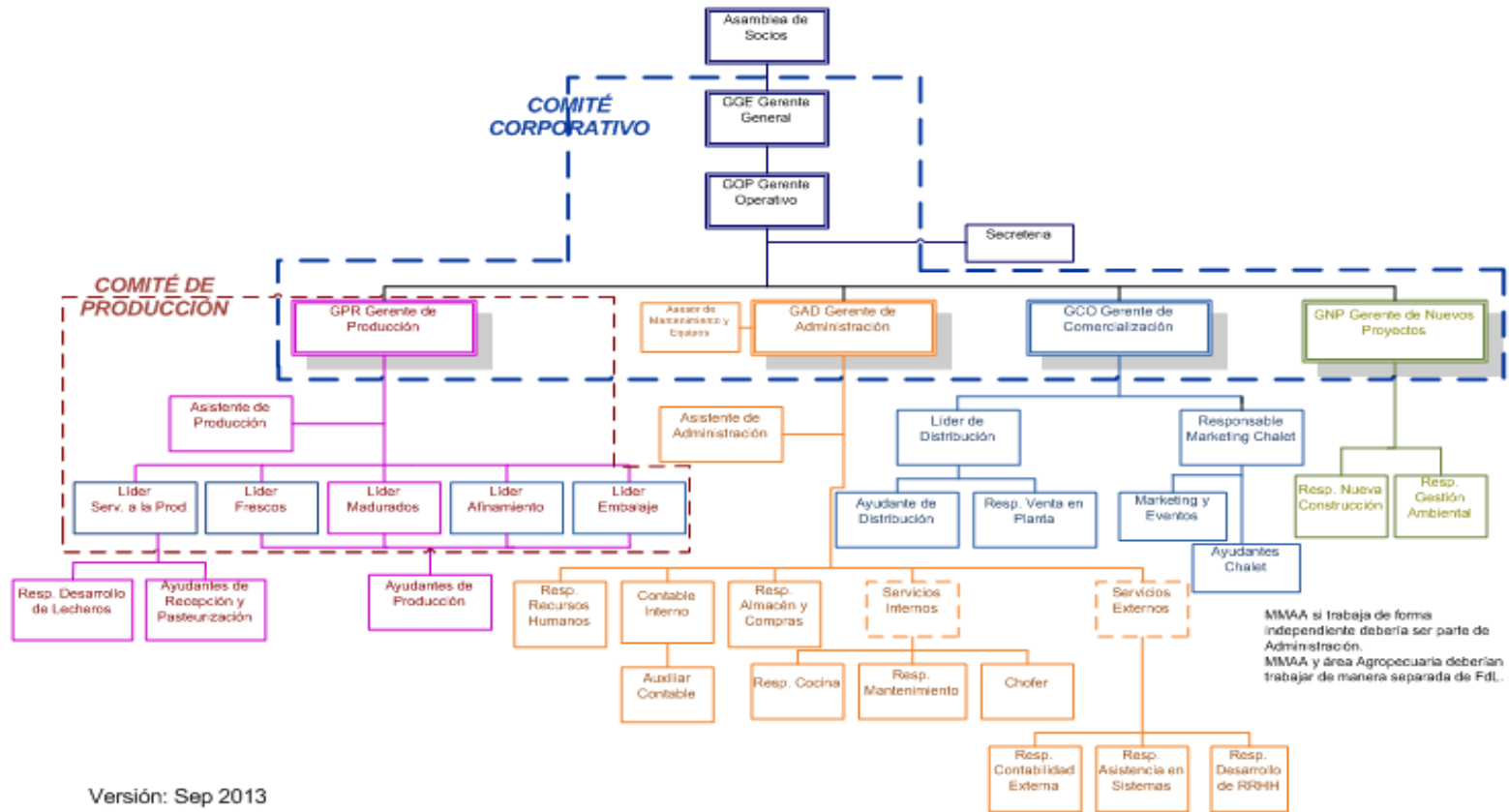
# ANEXOS



# ANEXO A



## ORGANIGRAMA FLOR DE LECHE S.R.L.



Versión: Sep 2013

## ANEXO B

Descripción de los casos de uso:

<b>Nombre:</b>	<b>Registrar empleados</b>	
<b>Actor:</b>	<b>Administración</b>	
<b>Descripción:</b>	<b>Describe el proceso de realizar el registro de empleados al sistema</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los documentos necesarios para realizar el registro	
	Realiza el registro del empleado	Realiza validación de datos necesarios
	Almacena los datos registrados	El sistema confirma el registro de datos
<b>Pre-condición</b>	El empleado es nuevo en la empresa	
<b>Post-condición</b>	El empleado se encuentra registrado.	
<b>Presunción</b>	Las base de datos de registro de empleados está disponible	

<b>Nombre:</b>	<b>Creación de grupos</b>	
<b>Actor:</b>	<b>Administración</b>	
<b>Descripción:</b>	<b>Describe el proceso de realizar la creación de nuevos grupos para los usuarios</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los grupos existentes	
	Realiza la creación del grupo.	Realiza validación del grupo creado y lo crea si pasa la validación
<b>Pre-condición</b>	El grupo es nuevo o no esta creado.	
<b>Post-condición</b>	El grupo es creado satisfactoriamente.	
<b>Presunción</b>	Las base de datos permite crear grupos de usuarios	

<b>Nombre:</b>	<b>Registrar Usuarios</b>	
<b>Actor:</b>	<b>Administración</b>	
<b>Descripción:</b>	<b>Describe el proceso de realizar la habilitación de un empleado como usuario de sistema</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los datos necesarios para ser usuario de sistema y escoge el grupo que le pertenece.	
	Realiza el registro del usuario	Realiza validación de datos necesarios
	Almacena los datos registrados	El sistema confirma el registro de datos
	Obtiene su login y password.	El sistema emite su login.
<b>Pre-condición</b>	El usuario es nuevo.	
<b>Post-condición</b>	El usuario se encuentra registrado y obtiene su login.	
<b>Presunción</b>	Las base de datos de registro de usuarios está disponible	

<b>Nombre:</b>	<b>Establecer roles</b>	
<b>Actor:</b>	<b>Administración</b>	
<b>Descripción:</b>	<b>Describe el proceso de realizar la administración de permisos de inserción, modificación, eliminación y ver información de la BD.</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los datos necesarios para dar permisos a los grupos existentes.	
	Establecer permisos.	Estable los permisos a los grupo asignados.
<b>Pre-condición</b>	El grupo existe en la BD.	
<b>Post-condición</b>	El grupo obtiene los permisos asignados por el administrador.	
<b>Presunción</b>	Las base de datos cuenta con grupos creados	

<b>Nombre:</b>	<b>Establecer personal de producción</b>	
<b>Actor:</b>	<b>Administración/GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de volver a los empleados parte del personal de producción.</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los datos necesarios para asignarlo parte del personal de producción	
	Registrarlo como empleado	Guarda la información en la BD y permite que aparezca en los responsables de fabricación.
<b>Pre-condición</b>	Existe el empleado en la BD.	
<b>Post-condición</b>	Ahora es responsable de fabricación.	
<b>Presunción</b>	Las base de datos cuenta con la tabla de personal de producción	

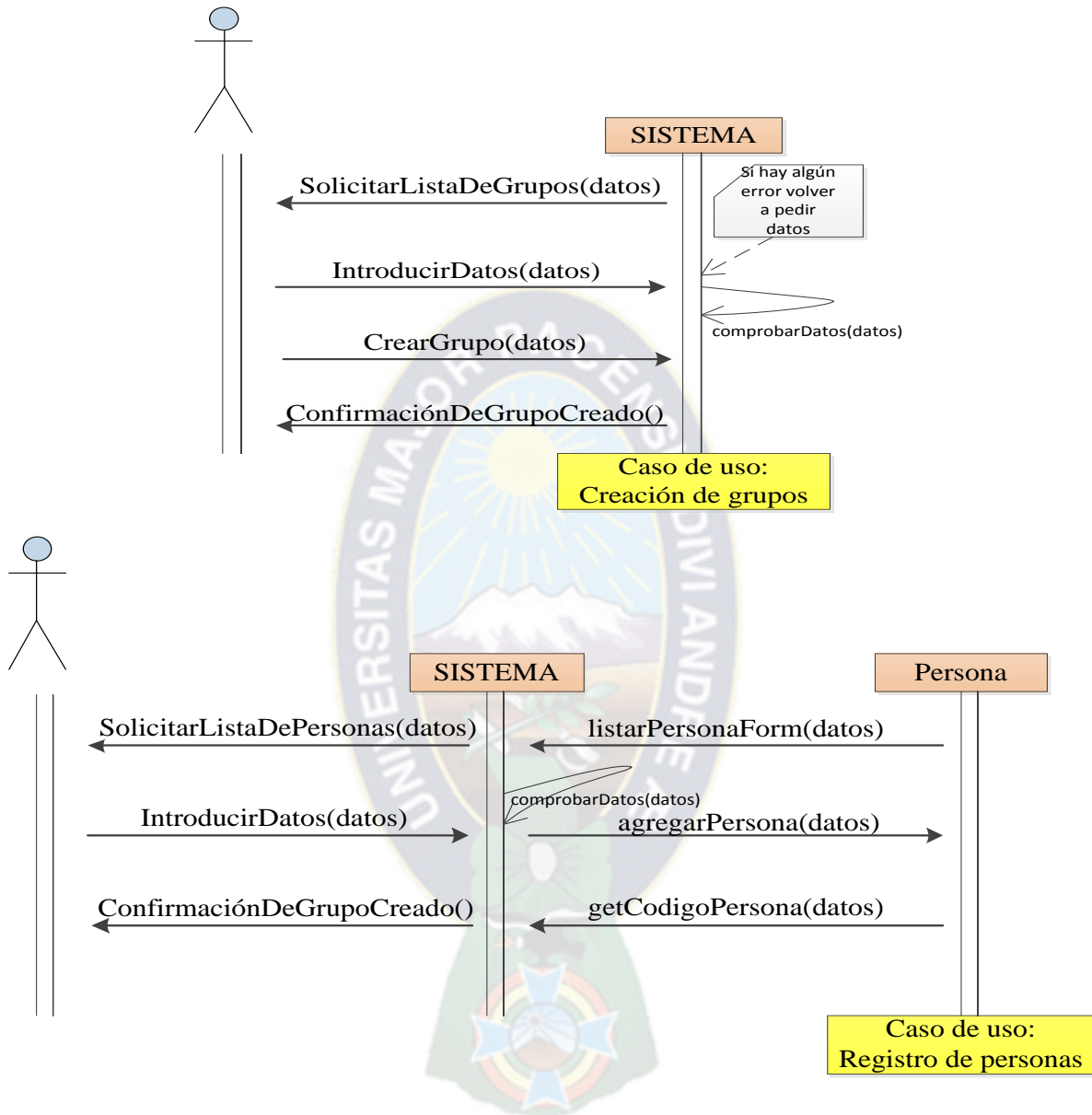
<b>Nombre:</b>	<b>Registrar tipos de producto</b>	
<b>Actor:</b>	<b>Administración/GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de registrar tipos de productos</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los tipos de productos existentes.	
	Registra nuevo tipo de producto.	Guarda la información en la BD y permite que aparezca en los productos.
	Se emite el código de tipos de producto	El sistema emite su código.
<b>Pre-condición</b>	No existe la descripción del tipo de producto.	
<b>Post-condición</b>	Inserción exitosa	
<b>Presunción</b>	Las base de datos cuenta con la tabla de tipos de producto	

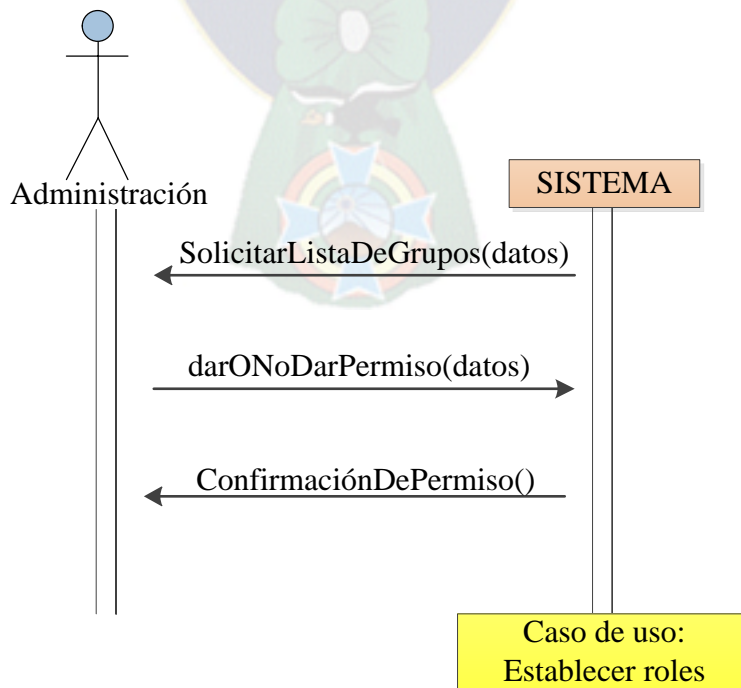
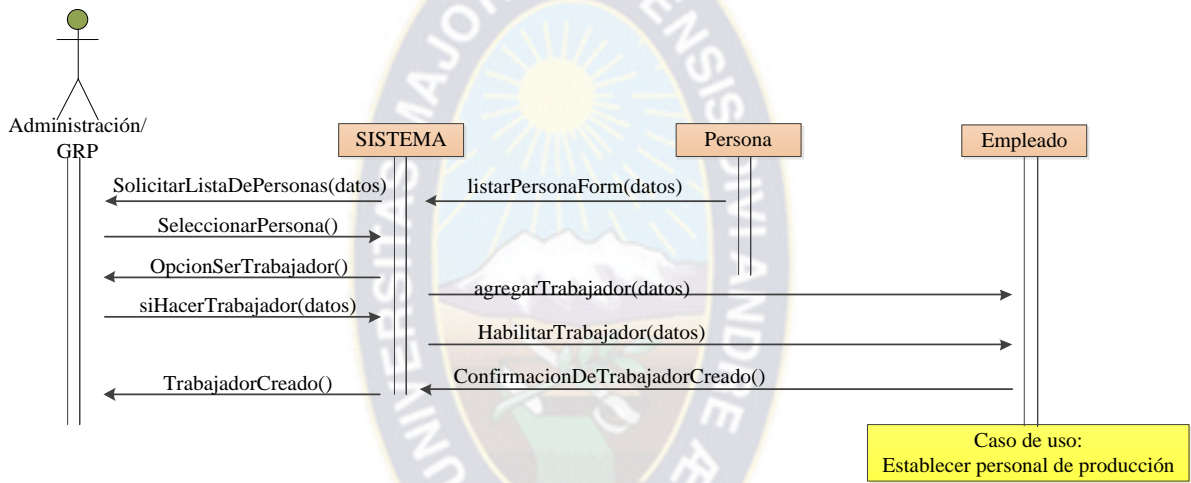
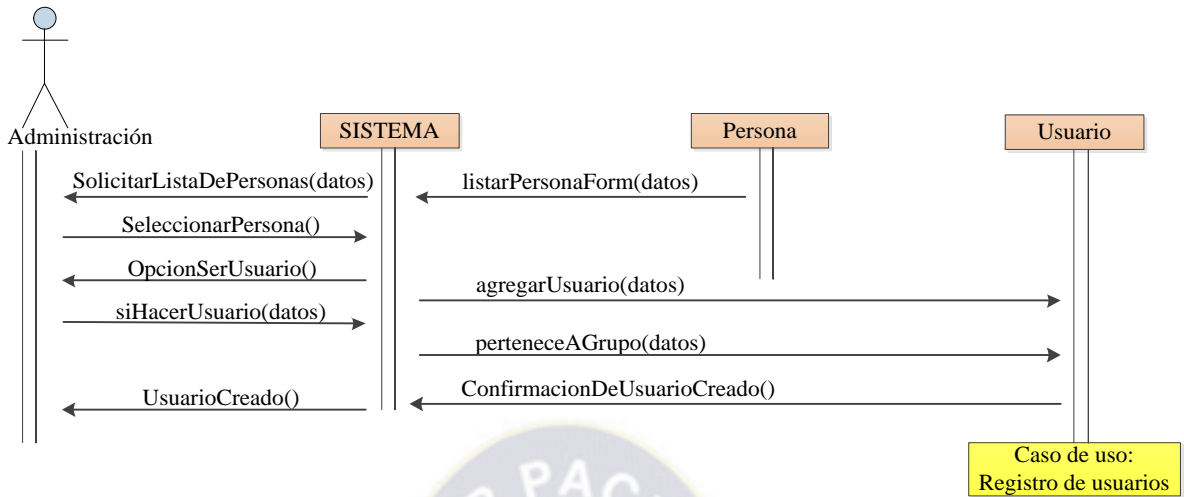
<b>Nombre:</b>	<b>Registrar productos</b>	
<b>Actor:</b>	<b>Administración/GPR(Gerencia de producción)</b>	
<b>Descripción:</b>	<b>Describe el proceso de registrar productos</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Verifica los productos y la información que se tiene del nuevo producto	
	Registrar nuevo producto.	Guarda la información en la BD.
	Se emite el código de producto.	El sistema emite su código.
<b>Pre-condición</b>	No existe la descripción del producto.	
<b>Post-condición</b>	Inserción exitosa	
<b>Presunción</b>	Las base de datos cuenta con la tabla productos	

<b>Nombre:</b>	<b>Entrega de reportes</b>	
<b>Actor:</b>	<b>GPR(Gerencia de producción)/Administración</b>	
<b>Descripción:</b>	<b>Describe el proceso de presentar o entregar reportes a Administración</b>	
<b>Flujo principal</b>	<b>Eventos del actor</b>	<b>Eventos del sistema</b>
	Colocar el nombre al reporte y exportarlos.	Guardar en la ruta especifica el reporte.
	Presentar al área de administración	
<b>Pre-condición</b>	Tener información seleccionada y validada.	
<b>Post-condición</b>	Gerencia conforme con el reporte.	
<b>Presunción</b>	GPR emite la información solicitada	

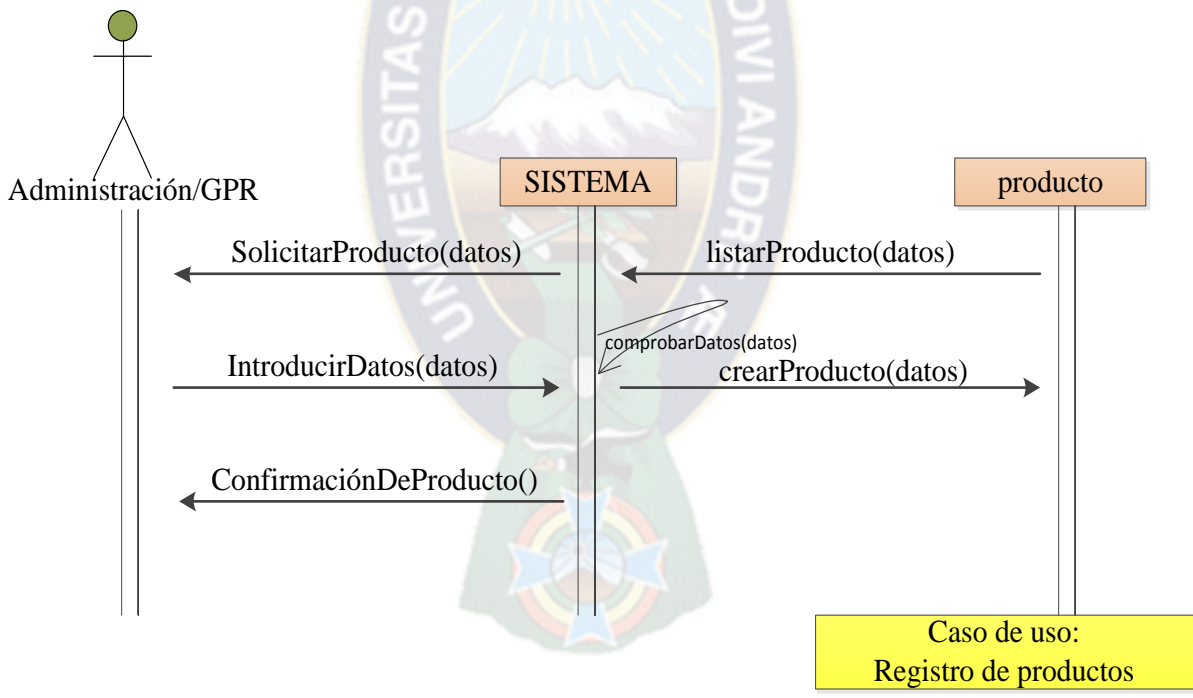
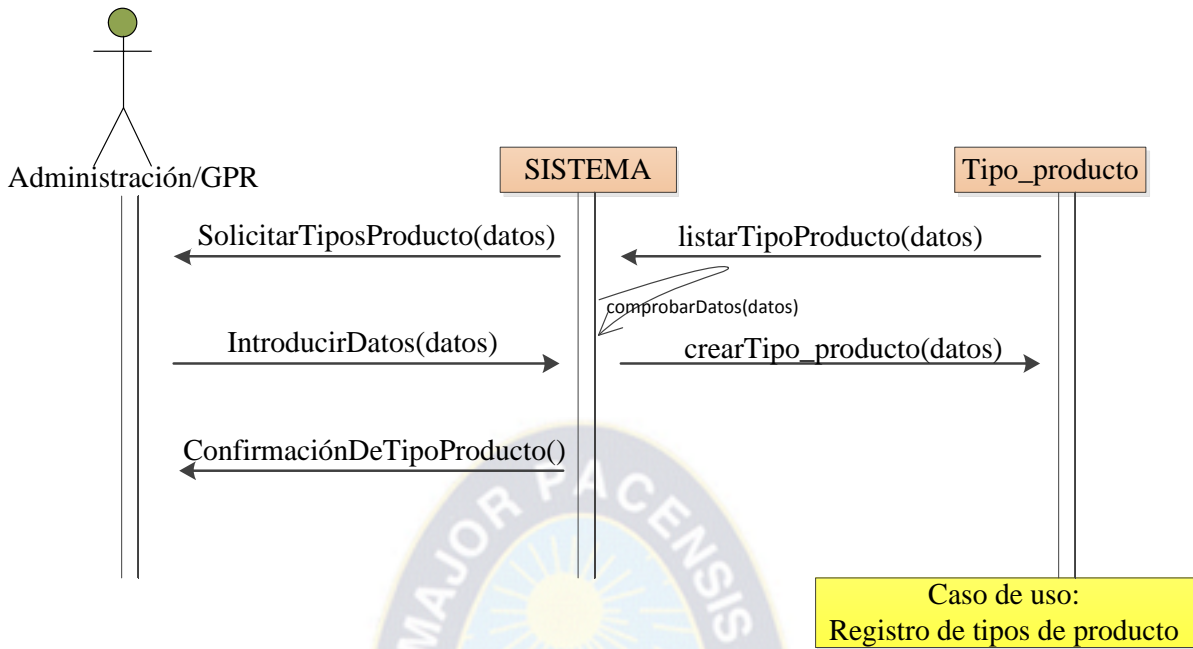
## ANEXO C

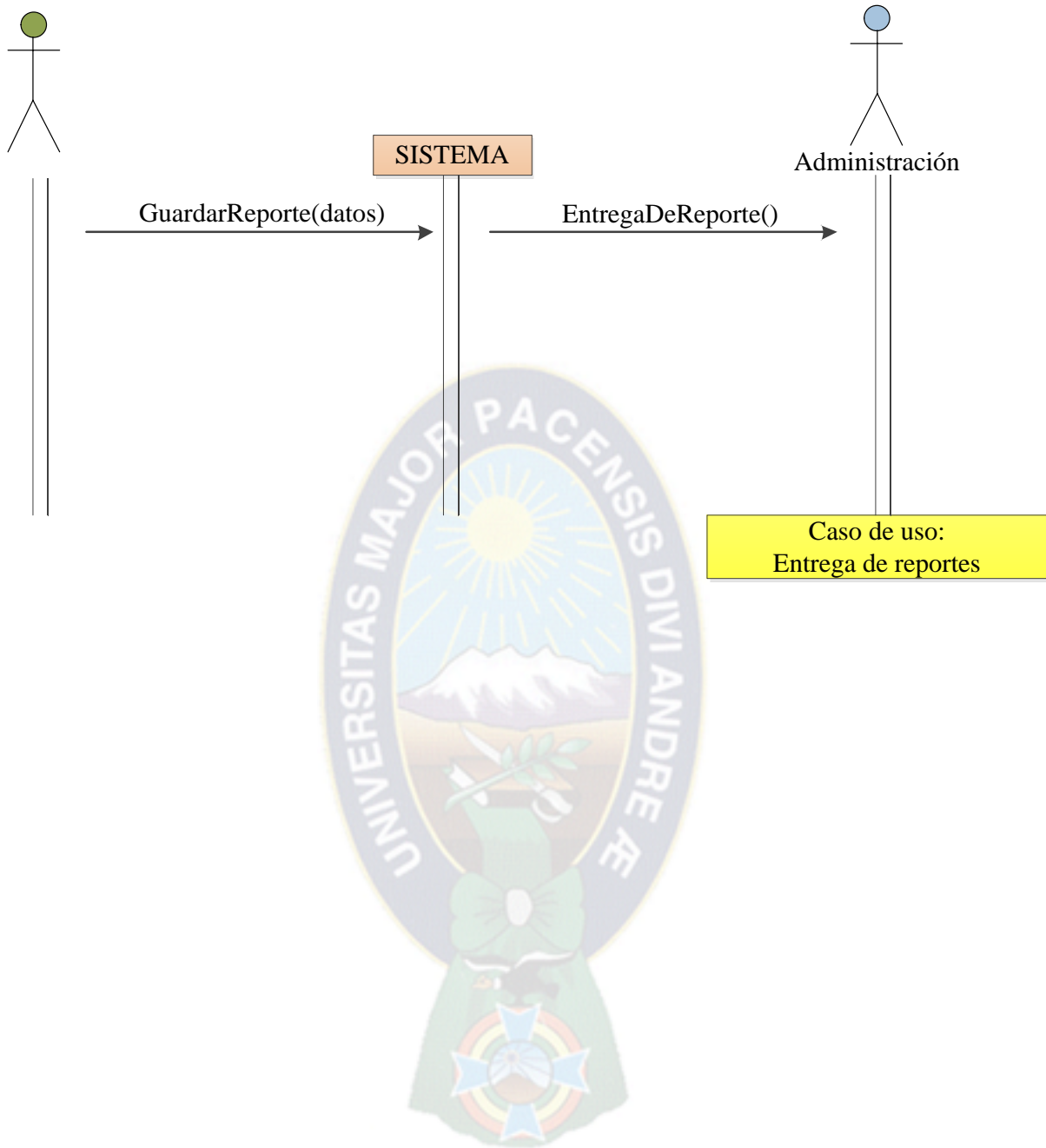
Diagramas de secuencia:











The watermark is a circular seal of the University of the Pacific. It features a sunburst at the top, a central emblem with a cross and a figure, and a banner at the bottom. The text "UNIVERSITAS MAJOR PACENSIS DIVI ANDREAE" is visible around the perimeter of the seal.

# DOCUMENTOS