



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**TEMA: SISTEMA DE CONTROL DE COSTOS DE  
PRODUCCIÓN DEL VIVERO DE PLANTAS NATIVAS Y  
HUERTOS AGROECOLÓGICOS, EN LA FUNDACIÓN  
CAMBUGÁN**

**AUTOR: MARCELO JAVIER MONTALVO VARGAS**

**TUTOR: ING. PAÚL VILLAVICENCIO**

**AÑO: 2016**

## RESUMEN

El proyecto tuvo como finalidad desarrollar un sistema para el control de los costos de producción de una Granja Agroecológica manejada por la Fundación Cambugán, en la cual se produce hortalizas y frutas agroecológicas, integrada con crianza de animales menores, entre otros productos naturales para la venta. Para proceder con el desarrollo del sistema se revisaron los procesos tanto de producción, como del cálculo del costo de cada producto y así tener la pauta para la construcción del mismo y poder cumplir el objetivo de determinar el costo real de los productos.

Xtreme Programming fue la metodología ágil de desarrollo de software utilizada para la construcción de este sistema, que sirvió de guía para el análisis, diseño, construcción y ejecución de las pruebas del mismo. Las herramientas de desarrollo que se utilizaron son de libre licencia, MySQL como motor de base de datos, PHP como lenguaje de programación, HTML, JQuery y Bootstrap para el front end del sistema. La arquitectura de la aplicación está basada en el patrón de diseño Modelo Vista Controlador que facilita el mantenimiento y orden de la programación.

La aplicación es en un entorno web y consta de 3 módulos principales: módulo de administración y mantenimiento del sistema (usuarios, productos, insumos), módulo de producción (creación de recetas de producción, amortización de inversiones órdenes de producción, planificación de producción) y módulo de reportes.

### **Palabras Clave.**

Sistema costos producción, metodología XP, aplicación web php, sistema producción agroecológica.

## **ABSTRACT**

The objective of the project was developed a system to control the production costs of an Agroecological Farm wich is managed by Cambugán Foundation. The farm produces agroecological vegetables and fruits, incorporating small animals in the raising system, and other natural products for their sale. The system was created with a survey of processes of agroecological production and the individual production cost of each product or of a production line that let us to determine the real cost.

Xtreme Programming is an agile development methodology to build software that is a guide for analysis, design, and construction for test execution. The development tools are open source. MySql an engine database, PHP programming language, HTML, JQuery and Bootstrap for the system front end. The architecture of the application is based on the design pattern Model View Controller, it makes easy the maintenance and scheduling order.

The application is in a web environment and has three main modules: Management Module and maintenance of the system (users, products and supplies), production module (creating recipes production, amortization investment, production orders, production planning) and reporting module.

### **Keywords.**

cost production system, XP methodology, php web application, agroecological production system.

## ÍNDICE DE CONTENIDOS

### **CAPITULO 1: INTRODUCCIÓN**

1.1 Antecedentes .....	1
1.2 Planteamiento del problema .....	2
1.3 Objetivos .....	3
1.3.1 Objetivo general .....	3
1.3.2 Objetivos específicos .....	3
1.4 Alcance .....	3

### **CAPITULO 2: MARCO TEÓRICO Y METODOLÓGICO**

2.1 Giro del negocio donde se va a aplicar la solución .....	4
2.1.1 La agroecología .....	4
2.1.2 Producción agroecológica .....	5
2.1.3 Costos de producción .....	6
Costo de producción Fundación Cambugán .....	6
2.2 Arquitectura de desarrollo .....	7
2.3. Herramientas de desarrollo .....	9
2.3.1 Selección motor de bases de datos .....	10
2.3.2 Selección del lenguaje de desarrollo .....	11
2.3.3 Selección de la metodología de desarrollo de software .....	13
2.4 Metodología ágil de desarrollo XP (Extreme Programming) .....	16
2.4.1 Valores de XP .....	16

2.4.2 Fases de XP .....	18
2.5 Técnicas de investigación aplicadas .....	19
2.5.1 La observación .....	19
2.5.2 La entrevista .....	21

### **CAPITULO 3: PRESENTACIÓN DE RESULTADOS**

3.1 Proceso de desarrollo de software.....	24
3.1.1 Los roles .....	24
3.1.2 Planificación .....	24
Historias de usuarios .....	24
Estimación de historias de usuario y plan de entregas .....	29
3.1.3 Diseño .....	29
Arquitectura .....	29
Especificación de escenarios (historias de usuario) .....	30
Diagrama de entidades base datos.....	37
Codificación .....	37
Cliente siempre disponible .....	37
Estándares de codificación .....	37
3.1.4 Pruebas .....	38
Pruebas de aceptación .....	38
Diseño de pruebas.....	38
3.2 Diseño de interfaces del sistema .....	54
CONCLUSIONES Y RECOMENDACIONES .....	60

Conclusiones .....	60
Recomendaciones .....	60
<b>BIBLIOGRAFÍA .....</b>	<b>62</b>
<b>ANEXOS .....</b>	<b>64</b>

## ÍNDICE DE TABLAS

Tabla 1. Cuadro comparativo Software libre vs Software Propietario .....	10
Tabla 2. Cuadro comparativo entre Bases de Datos - Software Libre .....	11
Tabla 3. Cuadro comparativo lenguajes de programación - Software libre .....	12
Tabla 4. Cuadro comparativo entre tipos de Metodologías de Desarrollo .....	14
Tabla 4.1. Cuadro comparativo entre Metodologías Agiles de Desarrollo .....	15
Tabla 5. Cuadro comparativo de sistemas de producción comerciales .....	21
Tabla 6.1. Historia de Usuario - Registro Usuarios .....	24
Tabla 6.2. Historia de Usuario - Registro Productos .....	25
Tabla 6.3. Historia de Usuario - Registro Insumos .....	25
Tabla 6.4. Historia de Usuario - Recetas de producción .....	26
Tabla 6.5. Historia de Usuario - Creación Ordenes de producción .....	26
Tabla 6.6. Historia de Usuario - Amortización de inversiones .....	27
Tabla 6.7. Historia de Usuario - Planificación de producción .....	27
Tabla 7. Estimación de Historias de usuarios .....	29
Tabla 8.1. Tarjeta CRC - Usuarios .....	30
Tabla 8.2. Tarjeta CRC - Productos .....	31
Tabla 8.3. Tarjeta CRC - Insumos .....	32
Tabla 8.4. Tarjeta CRC - Recetas .....	33
Tabla 8.5. Tarjeta CRC - Órdenes de producción .....	34
Tabla 8.6. Tarjeta CRC - Amortización de inversiones .....	35
Tabla 8.7. Tarjeta CRC - Planificación de Producción .....	36

Tabla 9.1. Prueba de aceptación - Ingreso de usuarios .....	39
Tabla 9.1.1 Prueba de aceptación - Edición de usuarios .....	40
Tabla 9.1.2 Prueba de aceptación - Eliminación de usuarios .....	41
Tabla 9.2. Prueba de aceptación - Ingreso de productos .....	42
Tabla 9.2.1. Prueba de aceptación - Edición de productos .....	43
Tabla 9.2.2. Prueba de aceptación - Eliminación de productos .....	44
Tabla 9.3. Prueba de aceptación - Ingreso de insumos .....	45
Tabla 9.3.1 Prueba de aceptación - Edición de insumos .....	46
Tabla 9.3.2 Prueba de aceptación - Eliminación de insumos .....	47
Tabla 9.4. Prueba de aceptación - Ingreso de recetas .....	48
Tabla 9.4.1. Prueba de aceptación - Edición de recetas .....	49
Tabla 9.4.2. Prueba de aceptación - Eliminación de recetas .....	50
Tabla 9.5. Prueba de aceptación - Crear Orden P. en estado En proceso .....	51
Tabla 9.5.1. Prueba de aceptación - Crear Orden P. para procesarla .....	52
Tabla 9.5.2. Prueba de aceptación - Cancelar orden de producción .....	53
Tabla 10. Detalle plan de entregas .....	65

## ÍNDICE DE FIGURAS

Figura 1. Cálculo del costo de producción Fundación Cambugán .....	7
Figura 2. Patrón de diseño MVC .....	8
Figura 3. Valores Metodología XP .....	16
Figura 4. Fases XP .....	18
Figura 5. Flujo de proceso de producción Fundación Cambugán .....	20
Figura 6. Arquitectura MVC del sistema .....	29
Figura 7. Diagrama de base de datos .....	37
Figura 8. Diseño de pruebas .....	38
Figura 9. Pantalla principal sistema .....	54
Figura 10. Listado de registros y acciones de mantenimiento de tabla .....	55
Figura 11. Acción de mantenimiento de tabla .....	56
Figura 12. Pantalla generar orden de producción .....	57
Figura 13. Crear receta de producción .....	58
Figura 14. Pantalla de reportes .....	59

## **Capítulo 1: Introducción**

### **1.1 Antecedentes**

La Fundación Cambugán, procura la conservación y restauración de ecosistemas boscosos, a través de la investigación científica y el trabajo comunitario. Nuestro objetivo es contribuir en la construcción de modelos de desarrollo sostenible que combinen la conservación del entorno natural y una vida digna de las comunidades relacionadas con él.

A partir del año 2011, la organización empieza la implementación del Proyecto Píntag, en un terreno de 9 hectáreas, con el fin de la construcción con un financiamiento limitado de una estructura física para la reproducción de árboles y plantas nativas, por lo que no se tuvieron los recursos suficientes para contratar los servicios de un técnico especializado en la sistematización de la información, lo cual no ha permitido mantener una cuantificación real de los costos, adicionalmente se tuvo problemas en la venta de los árboles, por el incumplimiento en tiempos de producción y la dificultad de estimación del precio de venta justo para los productos.

En el año 2013, se tomó la decisión de diversificar las actividades que se desarrollaba en el proyecto con la finalidad de crear un modelo de réplica de desarrollo sustentable del territorio en comunidades rurales, por lo que comienza la implementación de la Granja Agroecológica San Isidro, en la cual se produce hortalizas y frutas agroecológicas, crianza de animales menores, entre otros productos naturales para la venta. Referente a lo mencionado se tiene cuantificado algo de información durante los 3 años de implementación de la granja, pero se necesita contar con un análisis de los costos de producción que respondan a la lógica de funcionamiento de este tipo de sistemas productivos, y además se pueda ofrecer a la sociedad productos con un precio justo tanto para el productor como para el consumidor, y de esta forma poder tener un flujo de información adecuada, con datos que permita tomar decisiones.

## **1.2 Planteamiento del problema**

La Fundación Cambugán para el proyecto de la Graja Integral necesita definir su proceso de producción y de algunos datos que se requieran para poder planificar, evaluar y controlar la actividad.

Debido a la falta de información, la fundación ha mantenido un proceso de venta de productos con precios de venta al público impuestos a base de información de la competencia, lo cual ha generado un déficit de \$ 4430.32 dólares en los meses de (enero – mayo), del año 2015, por lo que se corre el riesgo de que el proyecto no pueda ser sostenible en los próximos años ocasionando que la fundación no pueda invertir en nuevos proyectos ni expandir su producción actual a otros niveles.

Actualmente la Fundación Cambugán, realiza de manera informal y sin documentación de sustento la planificación, evaluación y control de producción del vivero de plantas nativas y huerto orgánico razón por la cual es de suma importancia desarrollar un sistema que controle los costos de producción desde el inicio hasta el final para así obtener el costo y poder determinar un precio de venta al público real que justifique la inversión y permita obtener un desarrollo sustentable y sostenible en el tiempo, caso contrario existirá en el futuro un impacto en la rentabilidad del proyecto.

## **1.3 Objetivos**

### **1.3.1 Objetivo General**

Desarrollar un sistema de control de costos de producción para el vivero de plantas nativas y huertos agroecológicos en la Fundación Cambugán, con el fin de obtener un costo real de los productos.

### **1.3.2 Objetivos Específicos**

- Analizar los procesos administrativos y de gestión de la producción de la Fundación Cambugán.
- Diseñar un sistema utilizando una metodología de desarrollo ágil que permita orientar la construcción del mismo.
- Construir un sistema aplicando herramientas web basadas en software libre.
- Probar las funcionalidades operativas del sistema.

### **1.4 Alcance**

El alcance de este proyecto es el desarrollo de un sistema de control de costos de producción para el vivero de plantas nativas y huertos agroecológicos, en la Fundación Cambugán, con el fin de automatizar y mejorar los procesos de producción para obtener un costo real de los productos para la toma de decisiones en la organización.

La aplicación se va a conformar por los siguientes módulos:

- Módulo de administración del sistema.
- Módulo de producción.
- Módulo de reportes.

## **Capítulo 2: Marco teórico y metodológico**

Para dar solución al problema de la investigación se detalla cómo se aplicaron varios conceptos que intervienen en el proceso como: el giro del negocio en el cual se enfocó la solución, el campo o área que se trabajó, el flujo del proceso al cual se va aplicar la solución, técnicas de investigación que se emplearon en la investigación, técnicas y herramientas tecnológicas que se utilizaron para la construcción del sistema.

### **2.1 Giro del negocio donde se va a aplicar la solución**

#### **2.1.1 La agroecología**

Las innovaciones tecnológicas hoy en día son analizadas desde diversas ópticas, ya sea en función de sus efectos asociados al incremento de rendimientos, optimización de recursos o a la reducción de tiempos, aplicado a varios entornos.

Como una ciencia aplicada, la agroecología utiliza conceptos y principios ecológicos para el diseño y manejo de agroecosistemas sostenibles, donde los insumos externos se sustituyen por procesos naturales como la fertilidad natural del suelo y el control biológico. La agroecología saca el mayor provecho de los procesos naturales y de las interacciones positivas en las explotaciones agrícolas con el fin de reducir el uso de insumos externos y crear sistemas agrícolas más eficientes. Los principios agroecológicos usados en el diseño y el manejo de los agroecosistemas mejoran la biodiversidad funcional de los sistemas agrícolas que es esencial para el mantenimiento de procesos inmunes, metabólicos y reguladores, claves para la función del agroecosistema.

- Aumentar el reciclaje de biomasa, con miras a optimizar la descomposición de materia orgánica y el ciclo de nutrientes a través del tiempo.

- Proveer las condiciones de suelo más favorables para el crecimiento vegetal, en particular mediante el manejo de la materia orgánica y el mejoramiento de la actividad biológica del suelo.
- Fortalecer el "sistema inmunológico" de los sistemas agrícolas, mejorando la biodiversidad funcional los enemigos naturales, antagonistas, etc.
- Minimizar las pérdidas de energía, agua, nutrientes y recursos genéticos mejorando la conservación y regeneración de suelos, recursos hídricos y la diversidad biológica agrícola.
- Diversificar las especies y los recursos genéticos en el agroecosistema en el tiempo y el espacio a nivel de campo y del paisaje.

### 2.1.2 Producción agroecológica

Los elementos que intervienen en el proceso de control de costos de una producción agroecológica:

- **Producto terminado.** Producto listo para el consumidor final, que ha pasado por diversos estados e incluso se necesitan otros productos intermedios (los insumos) y materias primas para su producción.
- **Insumos.** Son todo lo necesario para producir el producto terminado. Si es que se necesita algún tipo de maquinaria o herramienta especial, estas también deben estar consideradas dentro de tu lista de insumos.

Los Insumos de producción pueden implicar un costo fijo o variable en el proceso de producción.

- **Receta de producción.** Es una lista de materiales o insumos que se necesita para producir un producto terminado en la cual se deben incluir las cantidades de cada insumo con sus respectivas unidades y en algunos casos el costo.

### 2.1.3 Costos de producción

"Los costos de producción, la productividad y el precio del producto son los tres factores más importantes en la determinación de la cantidad de productos o servicios que una organización está dispuesta a ofrecer en un mercado. Los costos se definen como el valor sacrificado para que una organización mantenga y consiga los recursos necesarios para la producción de bienes y servicios que necesariamente generan un beneficio futuro." (López, 2012)

#### Tipos de recursos para costos de producción

- **Recursos fijos:** "Conjunto de recursos cuya cantidad no puede ser modificada por la organización en un corto plazo, y que pueden soportar volúmenes diferentes de producción". (López, 2012)
- **Recursos variables:** "Conjunto de requerimientos necesarios para la producción que varía directamente en función de su volumen. Como son los insumos y materias primas, trabajadores, energía. Se sintetiza que los costos asociados a los recursos variables son: variables en cantidad y fijos por unidad". (López, 2012)

#### Costo de producción Fundación Cambugán

A continuación se detalla mediante un gráfico el flujo del proceso para el cálculo del costo de producción en la Fundación Cambugán.

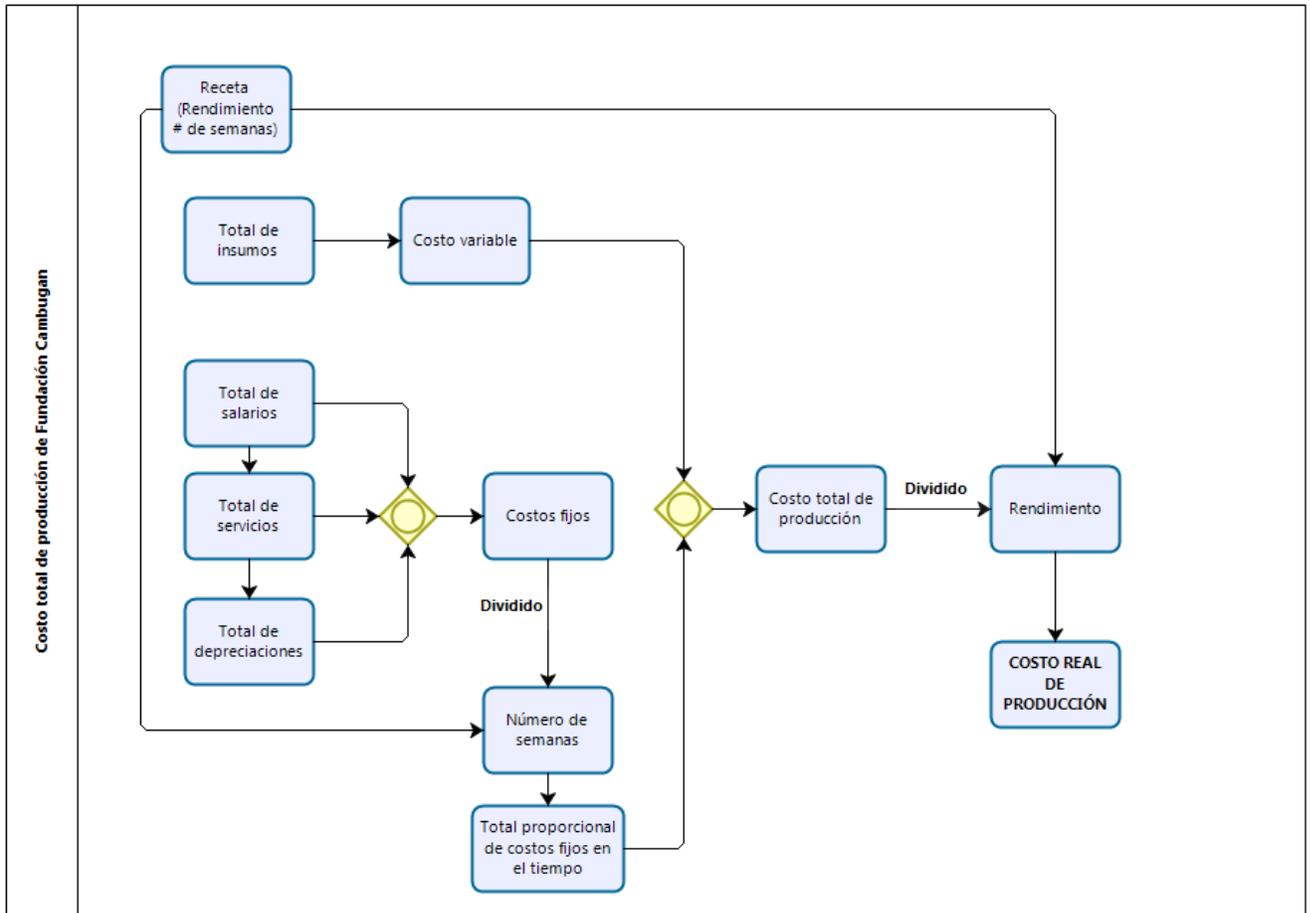


Figura 1: Flujo de proceso para el cálculo del costo de producción Fundación Cambugán

Fuente: por Marcelo Montalvo

## 2.2 Arquitectura de desarrollo

### Patrón de diseño MVC

"El patrón de diseño Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos dando como resultado una programación a nivel de 3 capas muy bien identificadas con su propia responsabilidad." (Alicante, s.f)

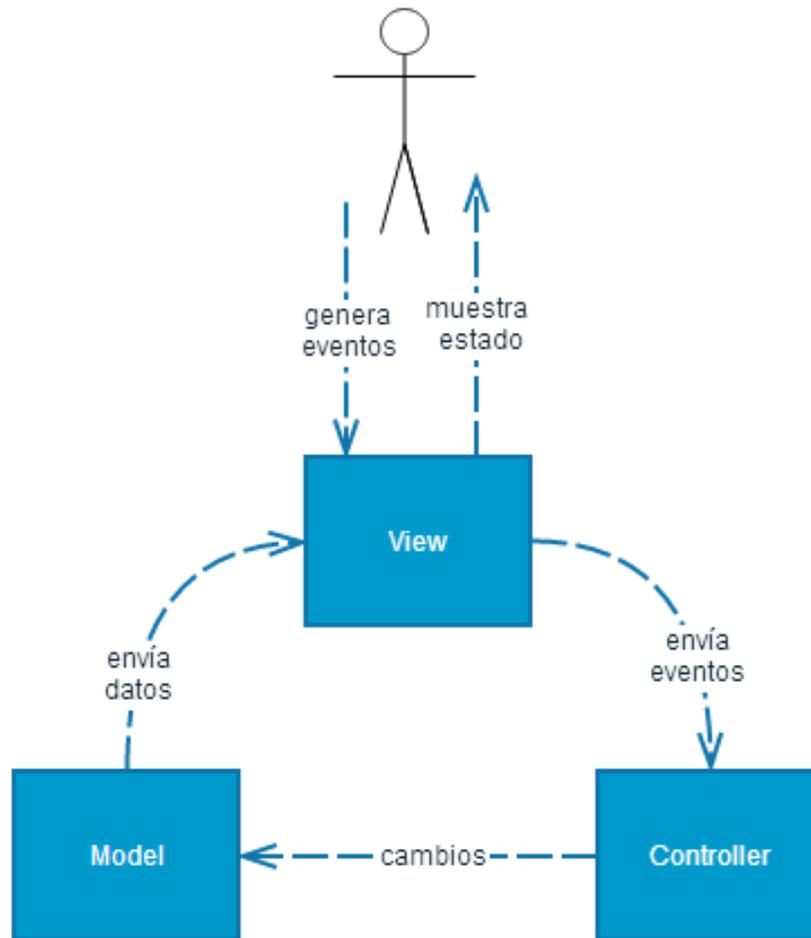


Figura 2. Patrón de diseño MVC

Fuente: <https://ingeniods.wordpress.com/2013/10/02/mvc-model-view-controller-patron-de-disenio/>

El modelo es el responsable de:

- "Acceder a la capa de almacenamiento de datos. Se modela las consultas de accesos a la base de datos tratándolos como objetos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento" (Alicante, s.f).

- "Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: (Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor)" (Alicante, s.f).
- "Lleva un registro de las vistas y controladores del sistema" (Alicante, s.f).

El controlador es responsable de:

- "Recibe los eventos de entrada ejecutados desde la aplicación (un clic, cambio en un campo de texto, etc.)" (Alicante, s.f).
- "Contiene reglas de gestión de eventos, del tipo (SI Evento Z, entonces Acción W). Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método Update(). Una petición al modelo puede ser Obtener\_tiempo\_de\_respuesta (nueva\_orden\_de\_compra)" (Alicante, s.f).

Las vistas son responsables de:

- "Recibir datos del modelo y los muestra al usuario" (Alicante, s.f).
- "Tienen un registro de su controlador asociado (normalmente porque además lo instancia)".
- "Brinda el servicio "actualizacion()", que es invocado por el controlador o por el modelo (Alicante, s.f)".

### **2.3 Herramientas de desarrollo**

Para la selección de las herramientas de desarrollo de software presentamos a continuación una tabla comparativa entre software libre y software propietario, la cual nos va ayudar a definir la orientación del tipo de herramientas a usar.

Tabla 1

Cuadro comparativo Software libre vs Software Propietario.

<b>SOFTWARE LIBRE</b>	<b>SOFTWARE PROPIETARIO</b>
No está limitado a los usuarios, y garantiza las libertades de los usuarios de usar, modificar, distribuir el software.	Existen restricciones por el motivo de licencias, las cuales son limitadas en número de usuarios. Estas licencias impiden la distribución del software.
Para su desarrollo puede intervenir cualquier persona, empresa u organización. Si bien se genera cualquier cantidad de ideas innovadoras, también se posibilita la oportunidad de adecuación de avances tecnológicos en estos productos.	El mantenimiento, desarrollo, programación y actualización del software solo lo hace la empresa que tiene los derechos.
Las actualizaciones y avances tecnológicos son constantes y se encuentran en la red de sin ningún costo. El objetivo del software libre es compartir los avances tecnológicos con la demás comunidad.	El software adquirido depende de una empresa comercial.
El usuario no depende de una sola empresa, ya que el software que implemento en su organización puede ser mantenido y modificado, acoplándolo a sus necesidades.	Existen renovaciones de licencias por estrategias comerciales. Siendo un costo para la empresa.

Fuente: <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-desoftware.html>

Se optó por el uso de herramientas de software libre para la construcción del sistema tanto en base de datos como en el lenguaje de programación.

### 2.3.1 Selección del Motor de Bases de datos.

Se evaluó diferentes motores de Bases de Datos, para determinar cual es la base de datos que más se acomodaba a las necesidades y a las exigencias de la aplicación que se desarrolló.

Entre las bases de datos más usadas en software libre se encuentran:

Postgres, MySQL y Firebird Database, escogiendo Mysql como el motor de base de datos ideal para la construcción de la aplicación, basándose en los indicadores de la tabla comparativa que se presenta a continuación:

Tabla 2

Cuadro comparativo entre Bases de Datos - Software Libre.

	Postgres	MySQL	Firebird Database
Mayor velocidad de transacción.	X	✓	X
Fácil integración con herramientas de desarrollo de Software.	✓	✓	✓
Consume pocos recursos de servidor.	X	✓	✓
Fácil integración de funcionalidades con PHP.	X	✓	X
<b>Resultados:</b>			
✓	1	4	2
x	3	0	2

Fuente: Por Marcelo Montalvo

### Características de MySQL:

- Es Open Source.
- La Velocidad en gestión y operaciones, hace que sea un motor de base de datos con alto rendimiento.
- Debido a su bajo consumo de recursos puede ser ejecutado en una máquina con escasos recursos de hardware.
- Facilidad de configuración e instalación.
- Puede ser instalada en variedad de Sistemas Operativos

### 2.3.2 Selección del lenguaje de desarrollo

Tabla 3

Cuadro comparativo lenguajes de programación - Software libre.

	<b>Java</b>	<b>PHP</b>	<b>Python</b>
Curva de aprendizaje rápida.	<b>X</b>	<b>✓</b>	<b>✓</b>
Comunidad de programadores alta.	<b>✓</b>	<b>✓</b>	<b>X</b>
Sencilla implementación del lenguaje en el servidor.	<b>X</b>	<b>✓</b>	<b>X</b>
<b>Resultados:</b>			
<b>✓</b>	<b>1</b>	<b>3</b>	<b>1</b>
<b>x</b>	<b>2</b>	<b>0</b>	<b>2</b>

Fuente: Por Marcelo Montalvo

Existen numerosas herramientas para el desarrollo de aplicaciones Web, entre las que se destacan: PHP, Java, y Python, en la Tabla 3 del documento se ha realizado un análisis comparativo con algunos indicadores entre los tres lenguajes de programación mencionados anteriormente, los cuales fueron un valor determinante en la selección del lenguaje más adecuado para el desarrollo de la aplicación.

Según los resultados se optó por PHP ya que es una herramienta de desarrollo web, confiable y de gran difusión con documentación extensa y es ideal para el desarrollo de aplicativos web sencillos y de gran complejidad.

A continuación alguna de sus características principales:

- Es un lenguaje de programación multiplataforma.
- Lenguaje de programación enfocado al desarrollo de aplicativos web dinámicos que interactúan con bases de datos.
- Integración con varios motores de base de datos que se utilizan en la actualidad, se destaca su conectividad con los motores PostgreSQL y MySQL.
- Su gran comunidad de PHP hace que el soporte, guías, libros y soluciones de dudas sea mucho más fácil en foros o redes sociales.
- PHP no requiere ningún tipo de licencia.
- Permite las técnicas de programación orientada a objetos a partir de la versión 5.

#### **2.3.4 Selección de la metodología de desarrollo de software**

Para la selección de una metodología de desarrollo de software presentamos esta a continuación una tabla comparativa, la cual nos ayudó a definir la opción con la que vamos a trabajar:

Tabla 4

Cuadro comparativo entre tipos de Metodologías de Desarrollo.

<b>Metodología Tradicional</b>	<b>Metodología Ágil</b>
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de Desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del Software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

Fuente: Por Marcelo Montalvo

Se optó por una metodología ágil ya que esta aumenta la productividad en el desarrollo, simplifica el manejo en la sobrecarga de los procesos y existe una mejor gestión del riesgo.

Existen numerosas metodologías ágiles para el desarrollo de software, aunque las tres metodologías más utilizadas son: Crystal Methodologies, Scrum, Xp (Extreme Programming), a continuación presentamos un cuadro comparativo que sirvió de ayuda para la selección de la misma:

Tabla 4.1

Cuadro comparativo entre Metodologías Ágiles de Desarrollo.

	<b>XP</b>	<b>Scrum</b>	<b>Crystal</b>
Enfoque de desarrollo.	Incremental iterativo.	Incremental iterativo.	Incremental iterativo.
Período de tiempo de iteración recomendada.	Una a 6 semanas.	Dos a cuatro semanas.	Lo define el Equipo.
Equipo de proyecto.	Equipos pequeños de menos de veinte miembros.	Todos los tamaños.	Todos los tamaños.
Documentación.	Básica.	Básica.	Básica.
Tamaño de proyectos.	Proyectos pequeños.	Todos los tipos de proyectos.	Todos los tipos de proyectos.
Implicación del cliente.	Cliente involucrado.	El cliente se involucra a través del rol de product owner.	El cliente se involucra a través de versiones incrementales.

Fuente: Por Marcelo Montalvo

XP se eligió como la metodología ágil entre otros motivos porque:

- Los requerimientos iniciales no fueron exactamente definidos los cuales cambiaron en el transcurso del proceso de desarrollo.
- Hubo una involucración directa del cliente en el transcurso del proyecto.
- El proyecto se basa en buenas prácticas las cuales fomentaron la comunicación, tanto cliente-desarrolladores como entre desarrolladores-desarrolladores, la sencillez, según fuera implementando el modulo se avanzaba con las necesidades para la construcción del mismo, realimentación, que ayuda en la corrección de errores para evitar complicaciones

a futuro, y la valentía para aceptar los cambios ya que los requerimientos variaron a lo largo del proyecto.

## 2.4 Metodología ágil de desarrollo XP (Extreme Programming)

"XP (Extreme Programming) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios." (Administrativa, 2006)

### 2.4.1 Valores de XP

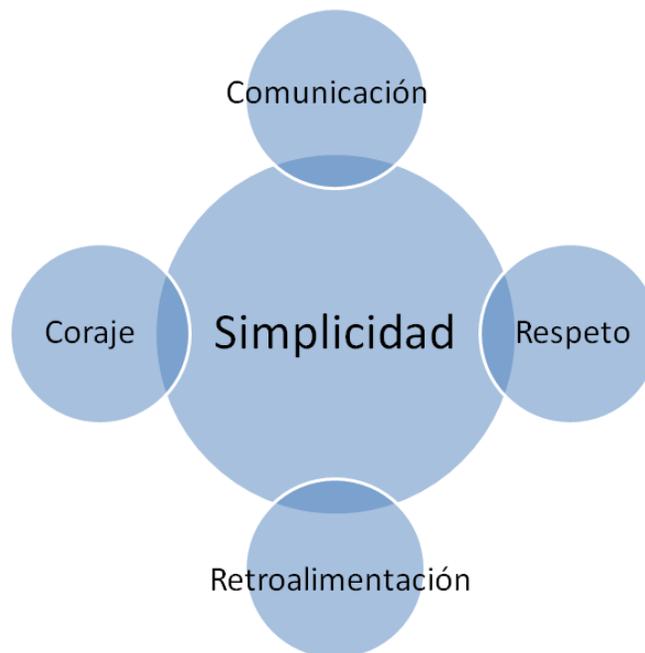


Figura 3: Valores Metodología XP

Fuente: Por [oficinasproyectosinformatica.blogspot.com](http://oficinasproyectosinformatica.blogspot.com)

Los valores en los que se fundamenta XP son 5. Estos valores son la guía para el desarrollo en sí mismo y la inspiración de toda la metodología.

**Simplicidad.-** Se desarrolla algo simple pero funcional cumpliendo siempre lo requerido por el cliente.

**Comunicación.-** Todos los involucrados participan en el proyecto, la comunicación de todas las partes es lo primordial con el fin de que no se desvíen los requerimientos.

**Realimentación.-** Se debe presentar al usuario el software frecuentemente y de forma temprana, aceptando todas las observaciones y realizando los cambios que sean necesarios. En sí es el feedback basado en varios puntos retroalimentación del sistema, retroalimentación del cliente y retroalimentación del equipo.

**Coraje.-** El trabajo es en equipo. El equipo debe adaptarse a los cambios cuando sea que estos se presenten.

**Respeto.-** El valor del respeto en la metodología promueve a que los aportes de cada integrante son valorados mutuamente por las partes involucradas en el proyecto.

La Metodología ágil XP expone ciertos criterios que indican el porqué se debe utilizar esta metodología para el desarrollo de software:

- Software funcional es mucho más importante que tener una documentación exhaustiva, si el software no es funcional la documentación no sirve de nada o esta mal.
- El interactuar con el cliente y su colaboración es más importante y productivo que una negociación de contratos.
- La realimentación permanente es la clave y éxito del proyecto.
- La respuesta y aceptación ante el cambio es más importante que el seguimiento de un plan.

## 2.4.2 Fases de XP



Figura 4: Fases XP

Fuente: Por Marcelo Montalvo

El ciclo de desarrollo de XP se puede dividir en 4 actividades fundamentales:

### **Planificación.**

Se recopilan y captan todos los requerimientos referentes al negocio.

Se exponen por parte del cliente y escriben las historias de usuario, y entre el cliente y los desarrolladores se agrupan (Por su valor) para priorizar los requerimientos que se van a hacer en la próxima versión.

En esta fase también se fijan tiempos estimados para la entrega.

### **Diseño.**

Se pone en práctica el realizar un diseño simple, para diseñar el producto de software la metodología XP propone las llamadas tarjetas CRC (Class - Responsibility - Collaborator) .

Estas tarjetas identifican y organizan las clases orientada a objetos que son relevantes para la versión.

### **Codificación.**

Los desarrolladores primero escriben las pruebas unitarias respecto a las historias de usuarios. Desarrolladas las pruebas, el programador está en condiciones de centrarse en lo que debe ser implementado para pasar la prueba. La programación en parejas es una parte clave en la fase de codificación ya que facilitará a una integración continua del código con el resto del equipo.

### **Pruebas.**

En la parte técnica se realizan las pruebas unitarias ya mencionadas anteriormente, adicional también existen las pruebas de aceptación (o pruebas de clientes) que son especificadas por el cliente centrándose en las características del sistema y en las funcionalidades visibles.

## **2.5. Técnicas de investigación aplicadas**

### **2.5.1 La observación**

Esta técnica fue aplicada en el proyecto para entender el giro del negocio donde se desarrolló el sistema. Se pudo constatar el flujo del proceso de producción en vivo para recoger información y de tal forma que ayude a la automatización de los procesos.

A continuación se detalla el flujo del proceso de producción de la Fundación Cambugán:

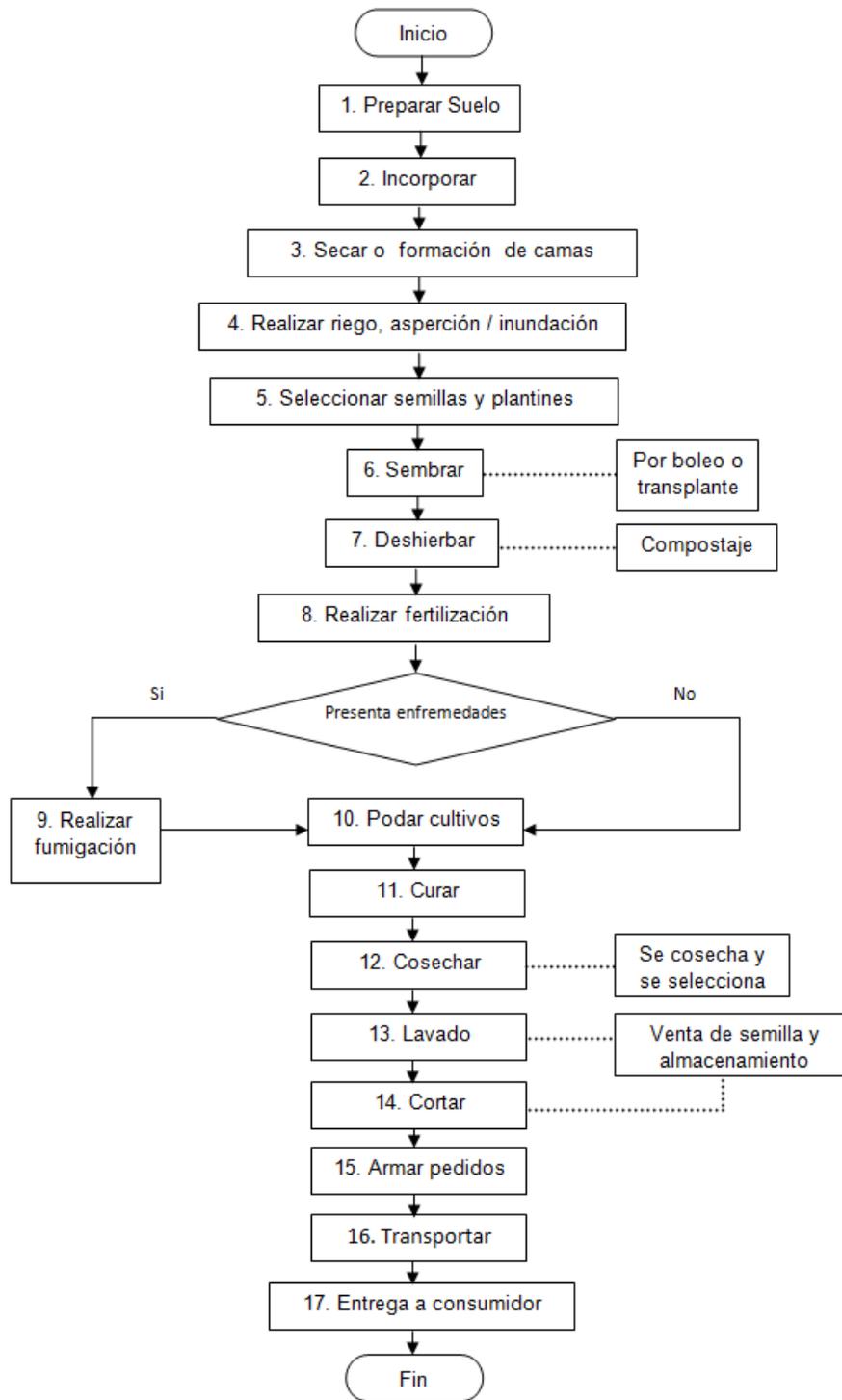


Figura 5: Flujo de proceso de producción Fundación Cambugán

Fuente: Por Marcelo Montalvo

## 2.5.2 La entrevista

La Entrevista se utilizó como técnica de investigación con el encargado del proceso de producción con el fin de captar los requerimientos para el diseño y desarrollo del sistema.

Se recopiló información de cómo lleva a cabo los cálculos de costos de producción, indicadores que se utilizan para llevar a cabo la producción, así como también herramientas que actualmente utilizan. Aplicando esta técnica se pudo evidenciar que el software es un desarrollo a la medida por lo que se realizó una comparación con otros productos de software enfocados a la temática a resolver en el proyecto con el fin de justificar el desarrollo de la aplicación.

Tabla 5

Cuadro comparativo de Sistemas de producción comerciales.

	<b>SynAgro (Sistema Agropecuario)</b>	<b>Agro Express (Gestión de Predios Agrícolas)</b>	<b>Sistema a la Medida</b>
Versión gratuita.	<b>X</b>	<b>X</b>	<b>✓</b>
Entorno web.	<b>X</b>	<b>X</b>	<b>✓</b>
Aplicación corre sobre un sistema operativo software libre.	<b>X</b>	<b>X</b>	<b>✓</b>
Módulos adicionales a producción.	<b>✓</b>	<b>✓</b>	<b>X</b>
<b>Resultados:</b>			
<b>✓</b>	<b>1</b>	<b>1</b>	<b>3</b>
<b>x</b>	<b>3</b>	<b>3</b>	<b>1</b>

Fuente: Por Marcelo Montalvo

Ninguna de las dos opciones de software mencionadas anteriormente se adapta totalmente a los requerimientos de la Fundación Cambugán, según la tabla comparativa los requerimientos técnicos son limitantes en los sistemas comerciales, adicionalmente existen requerimientos específicos en el proceso de producción que están atados a varios de los productos y servicios que brinda la Fundación por lo que se determinó que estos sistemas mencionados en la Tabla 5 tienen módulos genéricos para la producción y no son muy aplicables al proceso.

## Capítulo 3: Presentación de resultados

### 3.1 Proceso de Desarrollo de software

#### 3.1.1 Definición de Roles

Se debe tomar en cuenta que el desarrollador del proyecto es una persona, por ende los roles definidos para el uso de la metodología XP fueron ocupados por la misma, y en algunos casos por el asesor encargado del proyecto.

- Programador:

Marcelo Montalvo realizó el diseño y escribió el código del sistema.

- Cliente:

Los desarrolladores del proyecto y un integrante del comité de proyectos, escriben las historias de usuario y las pruebas funcionales para validar su implementación.

- Encargado de pruebas (Tester):

El integrante del comité de proyectos ayudó a escribir las pruebas funcionales. Ejecutó las pruebas regularmente, e informó los resultados y apreciaciones al equipo de desarrollo.

- Encargado de seguimiento (Tracker):

El integrante del comité proporcionó realimentación al equipo y realizó el seguimiento del progreso de cada iteración.

- Entrenador (Coach):

Marcelo Montalvo fue el responsable del proceso global. Fue el encargado de proveer guías al equipo de forma que se apliquen las prácticas XP y se siguiera el proceso correctamente.

Algunos roles definidos en la metodología XP como (Consultor y gestor o Big Boss) no fueron tomados en cuenta para este proyecto, por el tamaño tan reducido del equipo de trabajo (Rodríguez, 2015).

### 3.1.2 Planificación

#### Historias de usuarios.

A continuación se describen las características que el sistema debe poseer:

Tabla 6.1.

Historia de Usuario - Registro Usuarios

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 1	<b>Nombre:</b> Registro Usuarios
<b>Usuario:</b> usuario rol administrador / sysAdmin	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Media (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Bajo (Alto / Media / Bajo)
<b>Descripción:</b>	
El sistema debe tener una pantalla de login para ingresar al sistema, además se debe tener una administración de usuarios que comprende registro, edición y eliminación de los mismos.	
<b>Observaciones:</b>	
Los usuarios deben tener roles para controlar políticas de acceso al sistema. Adicional debe haber administración para crear varios tipos de roles a futuro.	
El módulo usuarios debe tener un estado de activo e inactivo para el control en el sistema.	

---

Tabla 6.2

Historia de Usuario - Registro Productos

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 2	<b>Nombre:</b> Registro Productos
<b>Usuario:</b> usuario rol administrado	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Bajo (Alto / Media / Bajo)
<b>Descripción:</b>	
Pantalla de ingreso, edición y eliminación de productos o servicios.	
<b>Observaciones:</b>	
Estos son los productos terminados para la producción.	

---

Tabla 6.3

Historia de Usuario - Registro Insumos

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 3	<b>Nombre:</b> Registro Insumos
<b>Usuario:</b> usuario rol administrado	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Bajo (Alto / Media / Bajo)
<b>Descripción:</b>	
Pantalla de ingreso, edición y eliminación de insumos.	
<b>Observaciones:</b>	
Son los elementos para la producción de un producto terminado.	

---

Tabla 6.4

Historia de Usuario - Recetas de producción

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 4	<b>Nombre:</b> Recetas de producción
<b>Usuario:</b> usuario rol administrado	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Bajo (Alto / Media / Bajo)
<b>Descripción:</b> Pantalla de ingreso, edición y eliminación de recetas.	
<b>Observaciones:</b> La receta para la producción es como un dato teórico en el cual se detalla los elementos compuestos y demás factores que intervienen en el proceso de producción de un producto terminado.	

---

Tabla 6.5

Historia de Usuario – Creación de órdenes de producción

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 5	<b>Nombre:</b> Creación de órdenes de producción
<b>Usuario:</b> usuario rol administrado	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Alta (Alto / Media / Bajo)
<b>Descripción:</b> Se van a crear órdenes de producción en la cuales se van a detallar todos los productos que se vayan a producir, desde la fecha de creación de la orden que representa el inicio de la producción hasta periodo de crecimiento de la receta del producto.	
<b>Observaciones:</b> El rendimiento para la producción es calculado en base a esta fórmula :	
rendimiento producto = superficie actual x rendimiento receta= / superficie receta	

---

Tabla 6.6

Historia de Usuario - Amortización de inversiones

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 6	<b>Nombre:</b> Amortización de inversiones
<b>Usuario:</b> usuario rol administrador / reportes	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Media (Alto / Media / Bajo)
<b>Descripción:</b>	
Pantalla con el listado de insumos que son inversión, que representa un costo fijo en la producción, se debe alertar el vencimiento de la amortización de los mismos.	
<b>Observaciones:</b>	
En el tablero principal se debe presentar una alerta, de que la inversión ya está vencida.	

---

Tabla 6.7

Historia de Usuario - Planificación de producción

---

<b>HISTORIA DE USUARIO</b>	
<b>Número:</b> 7	<b>Nombre:</b> Planificación de producción
<b>Usuario:</b> usuario rol administrador / reportes	
<b>Modificación de historia de usuario número:</b> 1	<b>Iteración asignada:</b> 1
<b>Prioridad de negocio:</b> Alta (Alta / Media / Baja)	<b>Riesgo en Desarrollo:</b> Alta (Alto / Media / Bajo)
<b>Descripción:</b>	
Pantalla con un esquema semanal en donde se visualiza los productos en proceso de producción que se crearon en las órdenes de producción. Existe la opción de parar una producción y calcular el costo de producción.	

## Observaciones:

Detalle de cómo se va a calcular el costo de producción:

Costos de producción:

- Se va a hacer una sumatoria de la multiplicación de costos unitarios de los insumos tipo costo variable por la cantidad que se encuentra en la receta para saber el costo variable del producto terminado a producirse.

$$\text{costo variable producto} = \sum (\text{costo unitario} \times \text{cantidad})$$

- Se va a hacer una sumatoria de los costos unitarios de los insumos tipo costo fijo, lo cual no va a dar un resultado de costo mensual. Como la planificación se la va a trabajar por semanas para la transformación a semanas aplicamos la siguiente fórmula:

$$\text{total costo fijo por semana} = (\sum \text{costo unitario} * 12) / 52$$

Nota: Para este cálculo existe una parametrización en el sistema en la sección parametrizaciones tabla 01 código PCF, donde en el dato2 se coloca el id del insumo de tipo costo fijo y en el dato1 se pone el porcentaje referente al costo total del insumo con el que se va a trabajar para la producción.

- Para el costo total del producto se va a realizar las siguientes acciones:
  1. El total costo fijo por semana hay que repartirlo por todos los productos que se estén produciendo en esa semana del año. Ej. Si mi costo fijo es 900 este monto se lo debe repartir para los 5 productos que se van a producir en la semana 6.
  2. El costo total final se lo calcula así:  $\text{costo variable producto} + \sum \text{proporcional total costo fijo por semana del producto}$  a calcular el costo

## Estimación de historias de usuarios y plan de entregas

Tabla 7

Estimación de Historias de usuarios

NRO.	HISTORIA DE USUARIO	TIEMPO ESTIMADO		
		SEMANAS	DIAS	HORAS
1	Registro de Usuarios	2	10	80
2	Registro de Productos	1	5	40
3	Registro de Insumos	1	5	40
4	Recetas de Producción	2	10	80
5	Creación de Órdenes de Producción	3	15	120
6	Amortización de Inversiones	3	15	120
7	Planificación de Producción	3	15	120
<b>Tiempo estimado total</b>		<b>15</b>	<b>75</b>	<b>600</b>

Fuente: Por Marcelo Montalvo

El plan de entregas se encuentra como anexo en la página No 64, tabla 10.

### 3.1.3 Diseño

En el diseño de la aplicación se evitó las soluciones complejas por lo que se va utilizar patrones de diseño para la construcción del sistema.

### Arquitectura

La construcción de la aplicación se basa en el patrón de diseño MVC (Modelo, vista, controlador) que se basa en construir la aplicación utilizando tres componentes, donde se diferencia cada uno con la separación de los datos, la lógica del negocio y la interfaz del usuario.

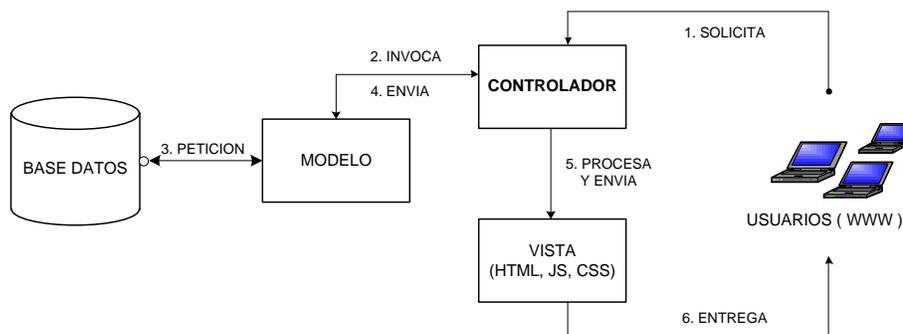


Figura 6. Arquitectura MVC del sistema. Por Marcelo Montalvo

## Especificación de escenarios (historias de usuario)

A continuación mostramos el diseño de nuestro sistema mediante las tarjetas CRC que representa los escenarios provenientes de las historias de usuario, los cuales especifican las posibles clases de la aplicación e identifican las responsabilidades y colaboraciones que son traducidas en métodos y relaciones en su respectiva iteración.

### ESCENARIO N° 1: REGISTRO DE USUARIOS

Propósito del Escenario:

1. Crear y registrar la información correspondiente al nuevo usuario.
2. Editar los datos de un usuario existente
3. Eliminar los datos de un usuario existente.

#### Tarjeta CRC: Usuarios

Tabla 8.1

Tarjeta CRC - Usuarios

TARJETA CRC		
<b>Numero:</b> 01	<b>Escenario:</b> Registro de usuarios	
<b>Nombre CRC:</b> Usuarios		
<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Registrar un nuevo usuario - Editar un Usuario	- Roles Usuario	add_nuevo_usuario edit_usuario
-Eliminar un usuario		delete_usuario
<b>Observaciones:</b>		
Se controla el registro y mantenimiento un usuario en el sistema. Los usuario se manejan por roles.		

## ESCENARIO N° 2: REGISTRO DE PRODUCTOS

Propósito del Escenario:

1. Registrar los productos terminados.
2. Editar la información de un producto existente
3. Eliminar un producto existente.

### Tarjeta CRC: Usuarios

Tabla 8.2.

Tarjeta CRC – Productos

---

### TARJETA CRC

---

**Escenario:** Registro de productos

**Numero:** 02

**Nombre CRC:** Productos

#### Responsabilidades

- Registrar un nuevo producto

- Editar un producto existente

-Eliminar un usuario

#### Colaboradores

- Categories

#### Métodos

add\_nuevo\_producto

edit\_producto

delete\_producto

#### Observaciones:

Se controla el registro y mantenimiento de productos en el sistema. Los productos van atados a una categoría.

---

## ESCENARIO N° 3: REGISTRO DE INSUMOS

Propósito del Escenario:

1. Registrar los insumos de producción.
2. Editar la información de los insumos
3. Eliminar un insumo existente.

### Tarjeta CRC: Usuarios

Tabla 8.3.

Tarjeta CRC – Insumos

---

### TARJETA CRC

---

**Escenario:** Registro de insumos

**Numero:** 03

**Nombre CRC:** Insumos

<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Registrar un nuevo insumo	- Categories	add_nuevo_insumo
- Editar un insumo existente		edit_insumo
- Eliminar un insumo		delete_insumo

#### **Observaciones:**

Se controla el registro y mantenimiento de insumos en el sistema. Los insumos van atados a una categoría.

Según el tipo de costo de insumo seleccionado se despliegan los detalles a ingresar.

---

## ESCENARIO N° 4: REGISTRO DE RECETAS

Propósito del Escenario:

1. Registrar una receta de producción.
2. Editar la información de la receta de producción.
3. Eliminar una receta existente.

### Tarjeta CRC: Usuarios

Tabla 8.4

Tarjeta CRC – Recetas

---

### TARJETA CRC

---

**Escenario:** Registro de recetas

**Numero:** 04

**Nombre CRC:** Recetas

<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Registrar una nuevo receta	- Productos	add_receta
- Editar una receta existente	- Insumos	edit_receta
-Eliminar un receta		delete_receta

### Observaciones:

Se controla el registro y mantenimiento de recetas en el sistema. En esta sección se añaden parametrizaciones de información necesaria para los productos terminados e insumos para crear las recetas.

---

## ESCENARIO N° 5: ÓRDENES DE PRODUCCIÓN

Propósito del Escenario:

1. Registrar una orden de producción.
2. Añadir detalles a una orden de producción.
2. Editar los detalles de una orden de producción.
3. Anular una orden de producción.

### Tarjeta CRC: Ordenes de Producción

Tabla 8.5.

Tarjeta CRC - Ordenes de producción

---

#### TARJETA CRC

---

**Escenario:** Registro de órdenes de producción

**Numero:** 05

**Nombre CRC:** Ordenes de producción

<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Crear una orden de producción	- Productos	open_orden
- añadir detalles a una receta de producción	- Recetas	anadir_detalle
- Cambiar estados a una orden	- Estados	edicion_detalle
		actualiza_estado_orden
		quita_detalle_orden
		get_lista_select_productos_receta
		lista_productos_recetas

**Observaciones:**

Se controla el registro y edición de órdenes de producción en el sistema. Las órdenes tienen estados en el proceso de producción. Dependencias de productos y recetas para crear una orden de producción.

## ESCENARIO N° 6: AMORTIZACIÓN DE INVERSIONES

Propósito del Escenario:

1. Mostrar lista de insumos de tipo inversión que tienen depreciación de costo fijo.
2. Mostrar desde su fecha de adquisición el valor de depreciación del bien respecto a la fecha actual.
3. Controlar si la inversión ha superado su fecha de vida útil.

### Tarjeta CRC: Amortización de inversiones

Tabla 8.6.

Tarjeta CRC - Amortización de inversiones

---

#### TARJETA CRC

---

**Escenario:** Amortización de inversiones

**Número:** 05

**Nombre CRC:** Amortización de inversiones

<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Mostrar Listado de inversiones de tipo costo fijo	- Insumos - Categorías	get_amortizacion_inversiones procesa_insumo_vencido
- cambiar a estado vencido		

#### **Observaciones:**

Se controla cuando el bien ya esta despreciado. Adicional se muestra alerta en Dashboard (pantalla principal) al usuario final.

## ESCENARIO N° 6: PLANIFICACIÓN DE PRODUCCIÓN

Propósito del Escenario:

1. Despliega en un esquema semanal partiendo de un rango de fechas todo lo que se encuentra en producción.
2. Existe la funcionalidad de parar la producción.
3. Muestra el costo de producción total por producto..

### Tarjeta CRC: Planificación de Producción

Tabla 8.7.

Tarjeta CRC - Planificación de Producción

---

<b>TARJETA CRC</b>		
<b>Número:</b> 06	<b>Escenario:</b> Planificación de Producción	
<b>Nombre CRC:</b> Planificación de Producción		
<b>Responsabilidades</b>	<b>Colaboradores</b>	<b>Métodos</b>
- Listar órdenes de producción desde un rango de fechas.	- Productos - Insumos - Recetas - Órdenes de producción	datos_produccion parada_produccion
- Actualizar en que semana termina la producción		costos costo_fijo_por semana
- mostrar detalle de costos del producto		
<b>Observaciones:</b> La presentación de datos es semanal. Lo que se encuentra en producción actual se muestra en la línea de tiempo de producción en color verde y lo ya producido con otro color.		

---

Otro aspecto importante aplicado en el diseño sin importar la fase del desarrollo, fue la constante refactorización del código y la puesta en práctica de los conceptos de una programación basada en una arquitectura Modelo Vista Controlador, que nos brindó el framework PHP de desarrollo con el que se trabajó, esto ayudó a generar en gran parte un código limpio con el fin de que este sea en un futuro entendible y mantenible.

## Diagrama de Base de datos

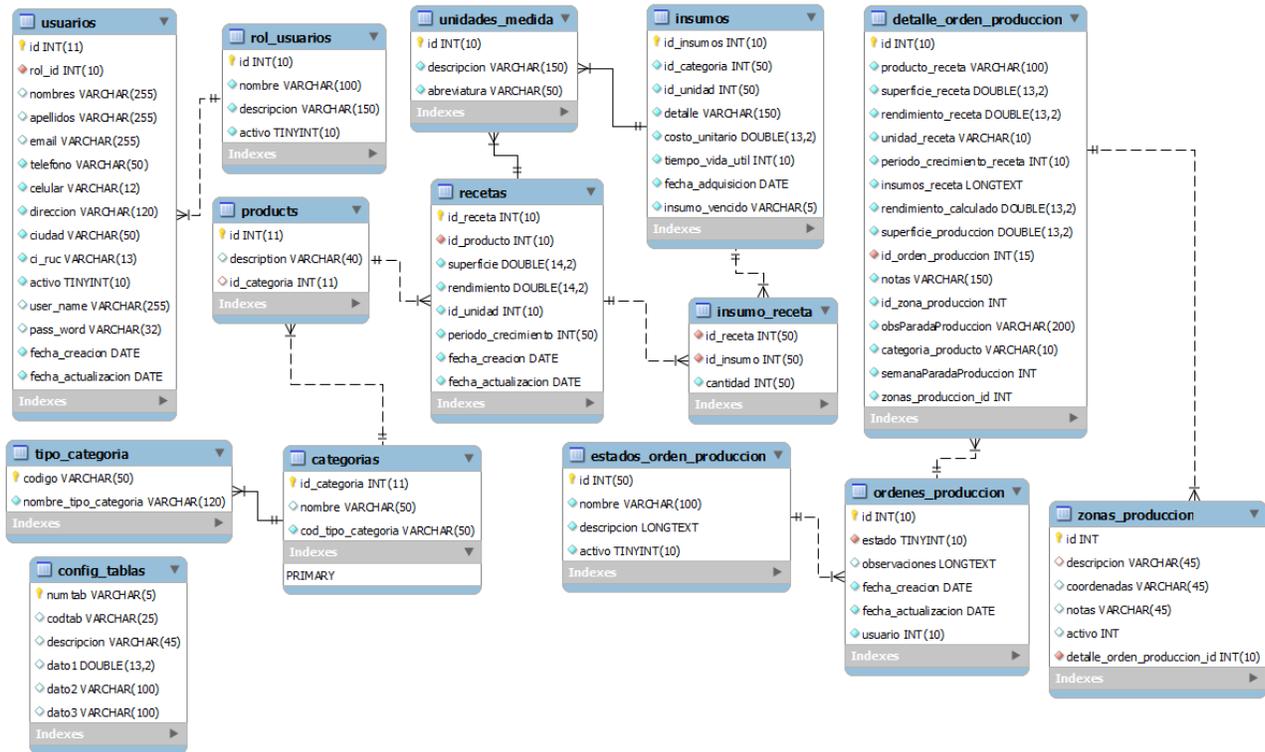


Figura 7: Diagrama Base de Datos

Fuente: Por Marcelo Montalvo

## Codificación

### Cliente siempre disponible

Como factor de éxito en la metodología XP es la involucración del cliente en todo el proceso de la construcción de software, este aspecto se cumplió a cabalidad ya que el cliente siempre estuvo disponible ante cualquier duda en los requerimientos y cambios que se realizaron en el transcurso del proceso.

### Estándares de codificación

La fijación de estándares es un aspecto muy importante en el desarrollo de una aplicación para que la construcción del proyecto sea uniforme y entendible por parte de cualquier otro integrante

del equipo de desarrollo. En el Anexo A se detallan los estándares que se aplicaron en el desarrollo.

### 3.1.4 Pruebas

#### Pruebas de aceptación

Mediante la planificación de iteraciones y en base a la especificación de historias de usuario, se crea las pruebas de aceptación, también denominadas pruebas de funcionalidad (documentos formales para determinar la aceptación de funcionalidad que posee un sistema (Historias de Usuario) ), las mismas que son constantes y constituyen uno de los pilares básicos de la metodología XP, permitiendo reducir el número de errores e incrementar la calidad del producto; representan una salida del sistema que el cliente espera sea funcional.

#### Diseño de Pruebas

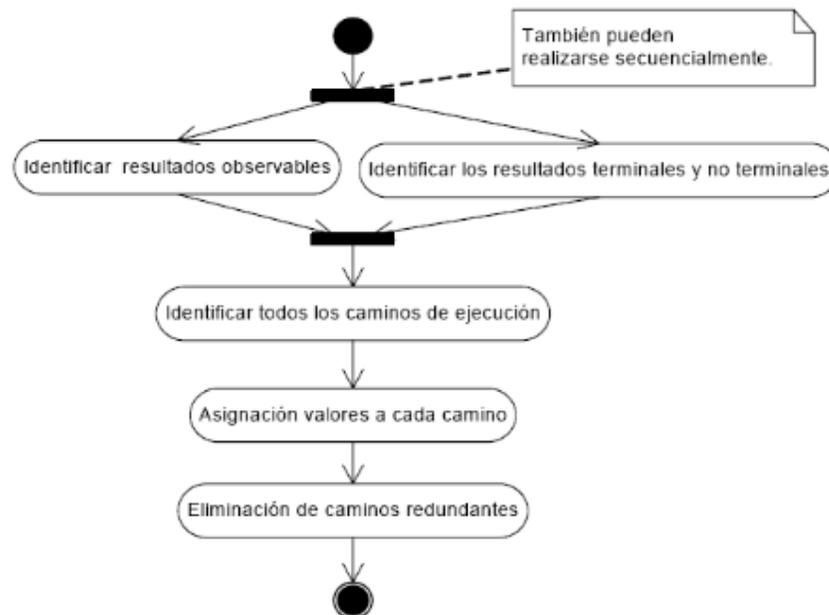


Figura 8: Diseño de pruebas

Fuente: GUTIÉRREZ J. J., ESCALONA M. J., MEJÍAS M., TORRES J. Pruebas del Sistema en Programación Extrema. University of Sevilla

## Escenario N° 1: Usuarios

Tabla 9.1

Prueba de aceptación - Ingreso de usuarios

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Usuarios

**Número Historia de Usuario:** 1

**Número caso de prueba:** 1

**Nombre caso de prueba:** Ingreso de usuarios

**Descripción:**

Se realiza el registro de la información del usuario con los campos que son obligatorios (Nombres, apellidos, nombre de usuario, contraseña, selección de rol, campo activo).

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios.

**Entradas**

1. Usuario accede como administrador opción usuarios.
2. El usuario da clic en el botón añadir nuevo
3. El usuario llena los campos que son obligatorios (Nombres, apellidos, nombre de usuario, contraseña, selección de rol, campo activo).
4. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro ingresado.
- El sistema presenta un mensaje de guardado con éxito.

**Evaluación**

El usuario es guardado correctamente.

---

Tabla 9.1.1

Prueba de aceptación - Edición de usuario

---

## PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Usuarios

**Número caso de prueba:** 1

**Número Historia de Usuario:** 1

**Nombre caso de prueba:** Edición de usuarios

### **Descripción:**

Se realiza la edición de la información del usuario con los campos que son obligatorios (Nombres, apellidos, nombre de usuario, contraseña, selección de rol , campo activo)

### **Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios.

### **Entradas**

1. Usuario accede como administrador opción usuarios.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al nombre del producto.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en el icono editar de la columna acciones para el usuario que corresponda.
6. El usuario llena los campos que son obligatorios (Nombres, apellidos, nombre de usuario, contraseña, selección de rol, campo activo).
7. El usuario pulsa el botón guardar cambios.

### **Resultado esperado**

- Se actualiza el registro seleccionado.
- El sistema presenta un mensaje de actualizado con éxito.

### **Evaluación**

El usuario es actualizado correctamente

---

Tabla 9.1.2

Prueba de aceptación - Eliminación de usuarios

---

## PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Usuarios

**Número caso de prueba:** 1

**Número Historia de Usuario:** 1

**Nombre caso de prueba:** Eliminación de usuarios

**Descripción:**

Se realiza la eliminación de la información del usuario.

**Condiciones de Ejecución:**

Lo usuarios deben estar ya registrado previamente.

**Entradas**

1. Usuario accede como administrador a la opción usuarios.
2. El usuario da clic en el botón filtrar.
3. El sistema despliega la información.
4. El usuario da clic en el icono eliminar de la columna acciones para el usuario que corresponda.
5. El usuario confirma la eliminación del registro.
6. El sistema presenta mensaje de eliminado correctamente.

**Resultado esperado**

- Se elimina el registro seleccionado.

**Evaluación**

El usuario es eliminado correctamente.

---

## Escenario N° 2: Productos

Tabla 9.2

Prueba de aceptación - Ingreso de productos

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Productos

**Número caso de prueba:** 1

**Número Historia de Usuario:** 2

**Nombre caso de prueba:** Ingreso de productos

**Descripción:**

Se realiza el registro de la información del producto con los campos que son obligatorios (Descripción y categoría).

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios.

**Entradas**

1. Usuario accede como administrador opción productos.
2. El usuario da clic en el botón añadir nuevo.
3. El usuario llena los campos que son obligatorios (Descripción y categoría).
4. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro ingresado.
- El sistema presenta un mensaje de guardado con éxito.

**Evaluación**

El producto es guardado correctamente.

---

Tabla 9.2.1

Prueba de aceptación - Edición de productos

---

## PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Productos

**Número Historia de Usuario:** 2

**Número caso de prueba:** 1

**Nombre caso de prueba:** Edición de productos

**Descripción:**

Se realiza la edición de la información del producto con los campos que son obligatorios (Descripción y categoría).

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios.

**Entradas**

1. Usuario accede como administrador a la opción productos.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al nombre del producto.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en el icono editar de la columna acciones para el producto que corresponda.
6. El usuario edita los campos que son obligatorios (Descripción y categoría).
7. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro seleccionado.
- El sistema presenta un mensaje de actualizado con éxito.

**Evaluación**

El producto es actualizado correctamente.

---

Tabla 9.2.2

Prueba de aceptación - Eliminación de productos

---

## PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Productos

**Número Historia de Usuario:** 2

**Número caso de prueba:** 1

**Nombre caso de prueba:** Eliminación de productos

**Descripción:**

Se realiza la eliminación de la información del producto.

**Condiciones de Ejecución:**

Lo productos deben estar ya registrados previamente.

**Entradas**

1. Usuario accede como administrador a la opción productos.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al nombre del producto.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en el icono eliminar de la columna acciones para el producto que corresponda.
6. El usuario confirma la eliminación del registro.

**Resultado esperado**

- Se elimina el registro seleccionado.

**Evaluación**

El producto es eliminado correctamente.

---

## Escenario N° 3: Insumos

Tabla 9.3

Prueba de aceptación - Ingreso de insumos

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Insumos

**Número caso de prueba:** 1

**Número Historia de Usuario:** 3

**Nombre caso de prueba:** Ingreso de insumos

**Descripción:**

Se realiza el registro de la información del insumo con los campos que son obligatorios según corresponda al tipo de insumo seleccionado.

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios. y creados los tipos de insumos.

**Entradas**

1. Usuario accede como administrador a la opción insumos.
2. El usuario da clic en el botón añadir nuevo.
3. El usuario llena los campos que son obligatorios si es un insumo tipo costo variable (Descripción, tipo insumo, unidad y costo unitario) y si es un insumo tipo costo fijo (Descripción, tipo insumo, unidad , costo unitario, tiempo vida útil, fecha adquisición).
4. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro ingresado.
- El sistema presenta un mensaje de guardado con éxito.

**Evaluación**

El insumo es guardado correctamente.

---

Tabla 9.3.1

Prueba de aceptación - Edición de insumos

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Insumos

**Número caso de prueba:** 1

**Número Historia de Usuario:** 3

**Nombre caso de prueba:** Edición de insumos

**Descripción:**

Se realiza la edición de la información del insumo con los campos que son obligatorios según corresponda al tipo de insumo seleccionado.

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios y creados los tipos de insumos.

**Entradas**

1. Usuario accede como administrador a la opción insumos.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al nombre del producto.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en editar de la columna acciones para el insumo que corresponda.
6. El usuario edita los campos obligatorios o no, si es un insumo tipo costo variable (Descripción, tipo insumo, unidad y costo unitario) y si es un insumo tipo costo fijo (Descripción, tipo insumo, unidad, costo unitario, tiempo vida útil, fecha adquisición).
7. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro editado.
- El sistema presenta un mensaje de actualizado con éxito.

**Evaluación**

El insumo es actualizado correctamente.

---

Tabla 9.3.2

Prueba de aceptación - Eliminación de insumos

---

**PRUEBA DE ACEPTACIÓN**

---

**Caso de prueba:** Insumos

**Número caso de prueba:** 1

**Número Historia de Usuario:** 3

**Nombre caso de prueba:** Eliminación de insumos

**Descripción:**

Se realiza la eliminación de la información del insumo.

**Condiciones de Ejecución:**

Lo insumos deben estar ya registrados previamente.

**Entradas**

1. Usuario accede como administrador a la opción insumos.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al nombre del insumo.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en el icono eliminar de la columna acciones para el insumo que corresponda.
6. El usuario confirma la eliminación del registro.

**Resultado esperado**

Se elimina el registro seleccionado.

**Evaluación**

El insumo es eliminado correctamente.

---

Tabla 9.4

Prueba de aceptación - Ingreso de recetas

---

**PRUEBA DE ACEPTACIÓN**

---

**Caso de prueba:** Recetas

**Número caso de prueba:** 1

**Número Historia de Usuario:** 4

**Nombre caso de prueba:** Ingreso de recetas

**Descripción:**

Se realiza el registro de la información de recetas con los campos (producto, superficie, rendimiento, unidad, periodo crecimiento, insumos).

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios y los insumos creados previamente.

**Entradas**

1. Usuario accede como administrador a la opción recetas.
2. El usuario da clic en el botón nueva receta.
3. El usuario llena los campos (producto, superficie, rendimiento, unidad, periodo crecimiento, insumos).
4. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro ingresado.
- El sistema presenta un mensaje de guardado con éxito.

**Evaluación**

La receta es guardada correctamente.

---

Tabla 9.4.1

Prueba de aceptación - Edición de recetas

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Recetas

**Número caso de prueba:** 1

**Número Historia de Usuario:** 4

**Nombre caso de prueba:** Edición de recetas

**Descripción:**

Se realiza la edición de la información de la receta de los campos (producto, superficie, rendimiento, unidad, periodo crecimiento, insumos).

**Condiciones de Ejecución:**

Deben estar llenos todos los campos obligatorios y los insumos creados previamente.

**Entradas**

1. Usuario accede como administrador a la opción recetas.
2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al código de la receta.
3. El usuario da clic en el botón filtrar.
4. El sistema despliega la información.
5. El usuario da clic en el icono editar en acciones para el insumo que corresponda.
6. El usuario edita los campos que son obligatorios o no, producto, superficie, rendimiento, unidad, periodo crecimiento, insumos.
7. El usuario pulsa el botón guardar cambios.

**Resultado esperado**

- Se actualiza el registro editado.
- El sistema presenta un mensaje de actualizado con éxito.

**Evaluación**

La receta es actualizada correctamente.

---

Tabla 9.4.2

Prueba de aceptación - Eliminación de recetas

---

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Caso de prueba:</b> Recetas	
<b>Número caso de prueba:</b> 1	<b>Número Historia de Usuario:</b> 4
<b>Nombre caso de prueba:</b> Eliminación de recetas.	
<b>Descripción:</b>	
Se realiza la eliminación de la información de la receta.	
<b>Condiciones de Ejecución:</b>	
Las recetas deben estar ya registrados previamente.	
<b>Entradas</b>	
<ol style="list-style-type: none"><li>1. Usuario accede como administrador a la opción recetas.</li><li>2. El usuario llena la caja de texto del buscador con los parámetros de búsqueda (una o varias letras) correspondientes al código de la receta.</li><li>3. El usuario da clic en el botón filtrar.</li><li>4. El sistema despliega la información.</li><li>5. El usuario da clic en el icono eliminar de la columna acciones para la receta que corresponda.</li><li>6. El usuario confirma la eliminación del registro.</li></ol>	
<b>Resultado esperado</b>	
<ul style="list-style-type: none"><li>• Se elimina el registro seleccionado.</li></ul>	
<b>Evaluación</b>	
La receta es eliminada correctamente.	

---

## Escenario N° 5: Órdenes de producción

Tabla 9.5

Prueba de aceptación - Crear Orden de producción en estado En proceso

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Órdenes de producción

**Número caso de prueba:** 1

**Número Historia de Usuario:** 5

**Nombre caso de prueba:** Crear orden de producción en estado En proceso

**Descripción:** Se realiza la creación de una orden de producción

**Condiciones de Ejecución:**

Recetas deben estar creadas previamente.

#### Entradas

1. Usuario accede como administrador a la opción órdenes de producción.
2. El usuario da clic en el botón nueva orden.
3. Automáticamente se crea la orden con fecha actual de creación y con estado en proceso.
4. El usuario añade los detalles de la orden seleccionado un producto e ingresando una superficie para que el campo rendimiento sea calculado automáticamente tomado los datos de la receta de ese producto.
5. El usuario pulsa el botón añadir detalle.
6. El usuario puede añadir N productos al detalle de la orden de producción.
7. El usuario puede editar el detalle ingresado cambiando, producto y superficie y el cálculo de rendimiento va a actualizarse.

#### Resultado esperado

- Se visualiza la orden en estado en proceso.
- El sistema presenta un mensaje de guardado con éxito.

#### Evaluación

La receta queda guardada en estado En proceso.

---

Tabla 9.5.1

Prueba de aceptación - Crear Orden de producción para procesarla.

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Órdenes de producción

**Número caso de prueba:** 1

**Número Historia de Usuario:** 5

**Nombre caso de prueba:** Crear orden de producción para procesarla

**Descripción:**

Se realiza la creación de una orden de producción.

**Condiciones de Ejecución:**

Recetas deben estar creadas previamente.

**Entradas**

1. Usuario accede como administrador a la opción órdenes producción, da clic en el botón nueva orden.
2. Automáticamente se crea la orden con la fecha de actual y con estado en proceso.
3. El usuario añade los detalles de la orden seleccionado un producto e ingresando una superficie para que el campo rendimiento sea calculado automáticamente.
4. El usuario pulsa el botón añadir detalle.
5. El usuario puede añadir N productos al detalle de la orden de producción.
6. El usuario puede editar el detalle ingresad, cambiando, producto, superficie y el cálculo de rendimiento va a actualizarse.
7. El usuario pulsa el botón procesar.

**Resultado esperado**

- Se guarda la orden de producción con estado Procesado y el sistema presenta un mensaje de guardado con éxito.

**Evaluación**

La orden queda guardada en estado procesado.

---

Tabla 9.5.2

Prueba de aceptación - Cancelar orden de producción

---

### PRUEBA DE ACEPTACIÓN

---

**Caso de prueba:** Órdenes de producción

**Número caso de prueba:** 1

**Número Historia de Usuario:** 5

**Nombre caso de prueba:** Cancelar una Orden de producción

**Descripción:**

Se abre una orden de producción con estado en proceso.

**Condiciones de Ejecución:**

Se debe tener órdenes de producción en estado en proceso.

**Entradas**

1. Usuario accede como administrador opción ordenes de producción.
2. El usuario busca la orden de producción.
3. El usuario da clic en el botón abrir de la columna acciones.
4. El usuario visualiza la orden de producción en esquema editable.
5. El usuario puede añadir N productos al detalle de la orden de producción.
6. El usuario pulsa el botón cancelar.
7. El sistema obliga al usuario a poner en el campo observación el porqué se va a cancelar la orden.

**Resultado esperado**

- Se actualiza el estado de la orden a cancelada.
- El sistema presenta un mensaje de actualizado con éxito.

**Evaluación**

La orden queda guardada en estado cancelado.

---

### 3.2 Diseño de interfaces del sistema

A continuación se muestra las pantallas prototipo de los módulos del sistema:

#### Pantalla principal (dashboard)

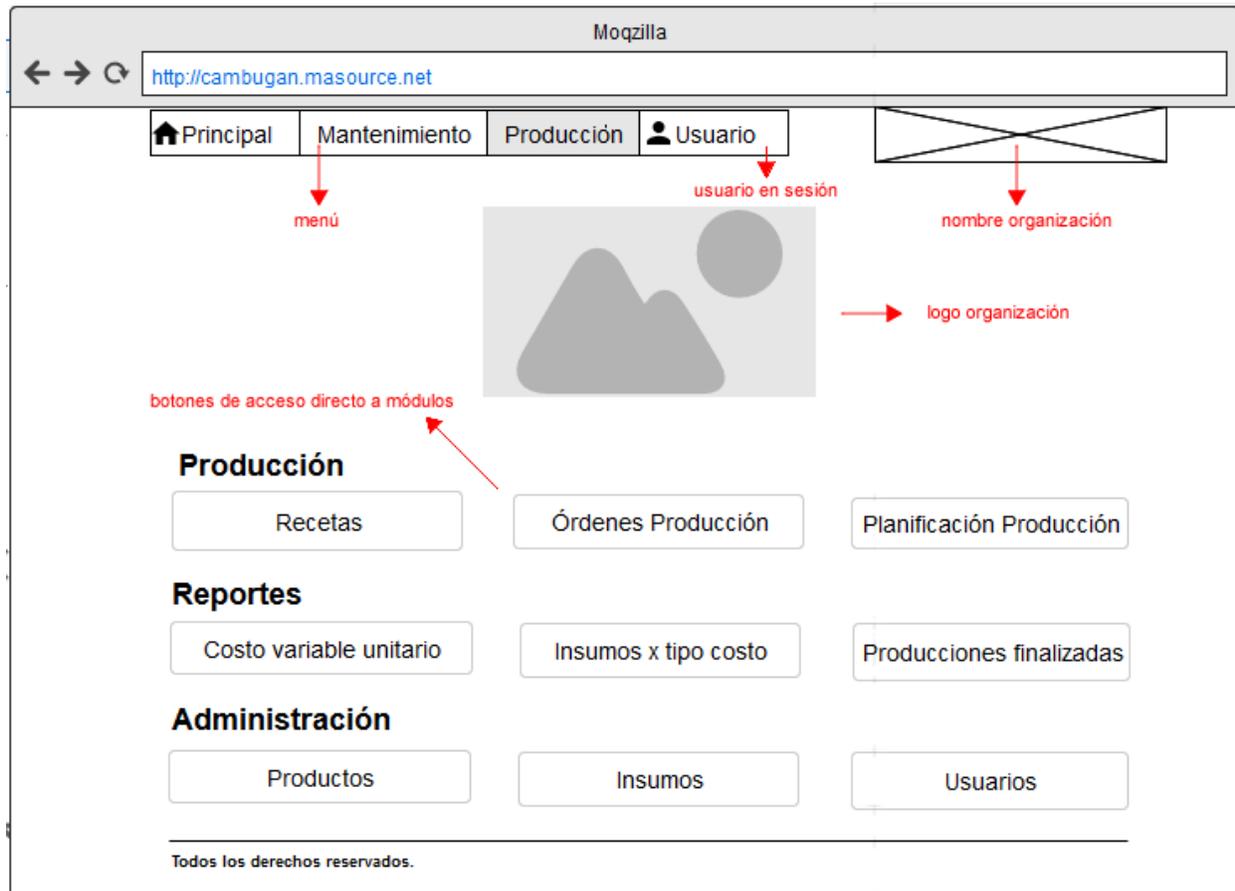


Figura 9: Pantalla principal de sistema (Dashboard).

Fuente: Elaborado por Marcelo Montalvo

#### Descripción:

En esta pantalla (Figura 9) se muestran todos los accesos y alertas que tiene el sistema. Existe un menú principal en la parte superior, un espacio para el logo de la organización, accesos directos a las opciones de los diferentes módulos del sistema, al final del menú se visualiza el usuario que inicia sesión en el sistema.

## Pantallas de mantenimiento del sistema (CRUD).

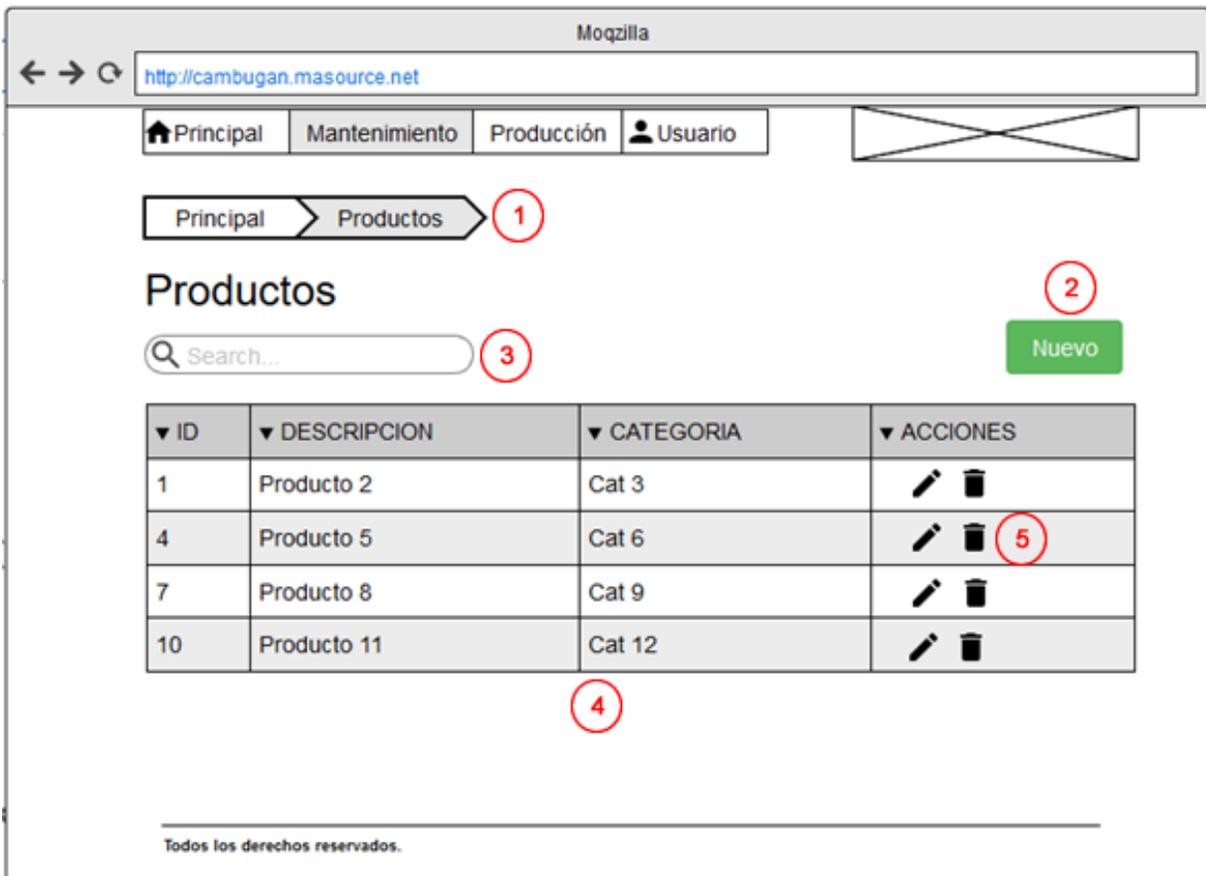


Figura 10: Listado de registros y acciones de un mantenimiento de tabla.

Fuente: Elaborado por Marcelo Montalvo

### Descripción(Figura 10):

1. Breadcrumbs (Navegación del sistema).
2. Botón para crear un nuevo registro.
3. Filtro para buscar en listado de registros.
4. Tabla con el listado de registros almacenados.
5. Acciones por registro: editar y eliminar.



Figura 11: Acción de mantenimiento de tabla.

Fuente: Elaborado por Marcelo Montalvo

En la Figura 11 se visualiza el diseño de la pantalla que se presenta luego de ejecutar una de las acciones de Figura 10. Esta pantalla es, tanto para crear un nuevo registro como editar un registro existente.

**Descripción:**

1. Breadcrumbs (Navegación del sistema).
2. Campos que se van a registrar en la tabla.
3. Botones de acción de la pantalla.

## Pantalla para generar una orden de producción



Figura 12: Pantalla generar una orden de producción.

Fuente: Elaborado por Marcelo Montalvo

### Descripción(Figura 12):

1. Breadcrumbs (Navegación del sistema).
2. Información orden de producción.
3. Campos para añadir un detalle (producto) a la orden de producción.
4. Lista de detalles de la orden de producción con sus respectivas acciones.
5. Botones de acción.
6. Icono que despliega detalle de receta de producción.

## Pantalla para crear una receta de producción

Mozilla

← → ↻ http://cambugan.masource.net

Principal Mantenimiento Producción Usuario

Principal Recetas (1)

### Recetas (2)

Producto Seleccione una ▼

Superficie

Rendimiento  (3)

Unidad

Período crecimiento

▼ Insumo	▼ Cantidad	▼ Unidad	
insumo 1	<input type="text"/>	lt	<input type="checkbox"/>
insumo 2	<input type="text"/>	lbs	<input type="checkbox"/>
insumo 3	<input type="text"/>	lbs	<input type="checkbox"/>
insumo 4	<input type="text"/>	gr	<input type="checkbox"/>

Guardar Salir (4)

Todos los derechos reservados.

Figura 13: Crear receta de producción

Fuente: Elaborado por Marcelo Montalvo

### Descripción (Figura 13):

1. Breadcrumbs (Navegación del sistema).
2. Listado de ingredientes (insumos) que componen el producto terminado.
3. Datos del producto terminado que se va a crear la receta.
4. Botones de acción.

## Pantalla de reportes

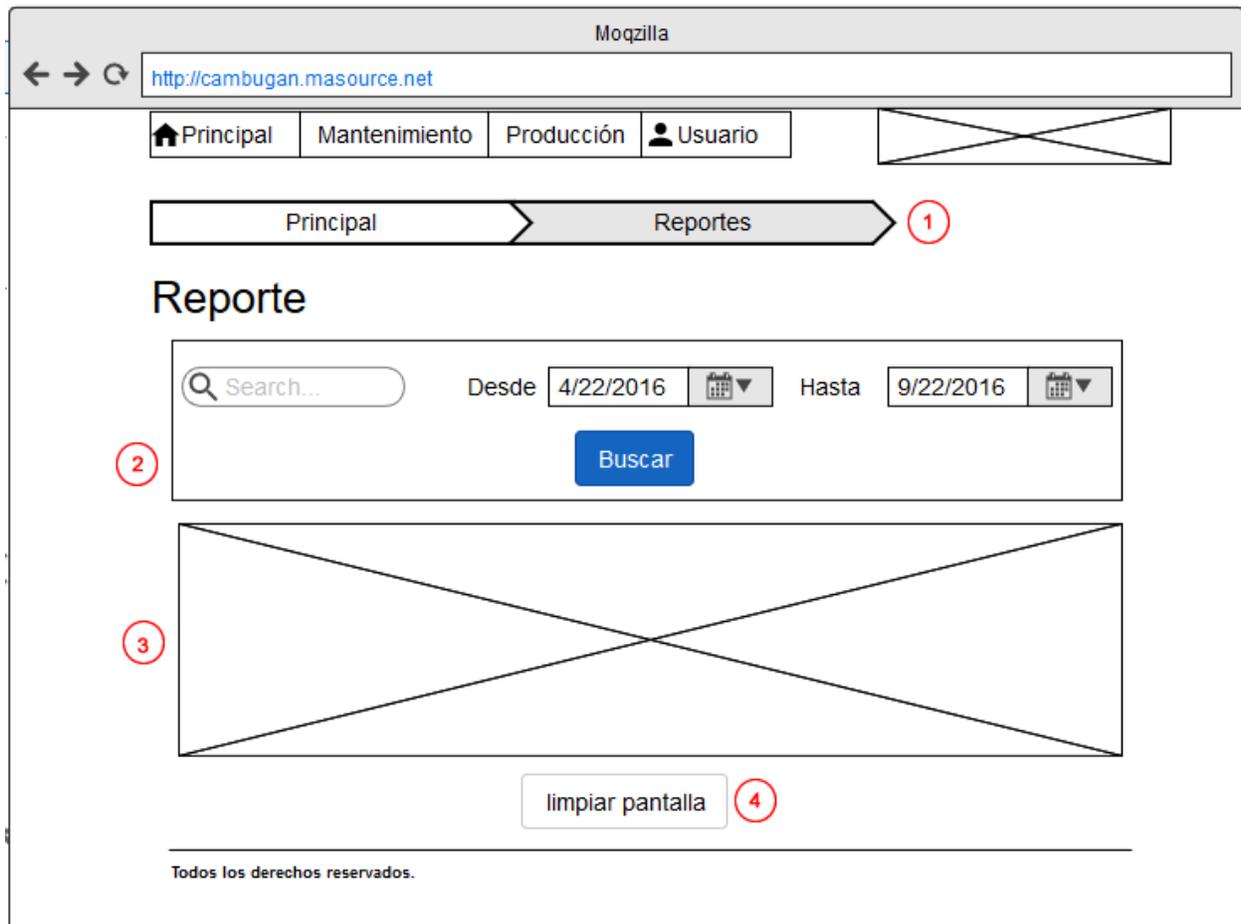


Figura 14: Pantalla de reportes.

Fuente: Elaborado por Marcelo Montalvo

### Descripción (Figura 14):

1. Breadcrumbs (Navegación del sistema).
2. Filtros para reportes.
3. Espacio para contenido de reporte.
4. Botón de acción.

## **CONCLUSIONES Y RECOMENDACIONES**

### **Conclusiones**

- Se realizó una revisión y análisis de los procesos que intervienen en el cálculo del costo de producción en la Fundación Cambugan, lo cual facilitó con mayor eficiencia la obtención de la forma de como calcular el costo para los productos.
- El contacto con los usuarios finales en el transcurso del desarrollo del sistema sirvió de ayuda para levantar los requisitos y así cumplir con sus necesidades y requerimientos en la automatización de los procesos.
- Se optó por una arquitectura web basada en el patrón de diseño MVC ( Modelo - Vista - Controlador ) para la construcción del sistema con el fin de que el software sea escalable y mantenible a futuro.

### **Recomendaciones**

- Establecer un plan automático de copias de seguridad de los archivos y datos que genera el sistema con el fin de mantener la información segura y respaldada frente a cualquier tipo de inconveniente que se presente.
- Realizar una capacitación dirigida a los usuarios finales del sistema para efectuar una revisión del manual de usuario, el cual sirve de guía para el uso correcto de las funcionalidades del sistema.
- Se recomienda utilizar la metodología XP en caso de realizar futuros desarrollos sobre el sistema, ya que esta nos facilita a identificar, comprender y realizar cambios en los requerimientos en cualquier fase del desarrollo.

- A corto plazo se recomienda desarrollar el módulo de ventas el cual trabajaría en conjunto con el módulo de costos de producción, ya que al saber el costo real de cada producto se puede calcular el PVP del mismo para la venta.

## BIBLIOGRAFÍA

- Administrativa, C. y. (2006, 01 15). *Ciencia y Tecnica Administrativa*. Retrieved from  
Métodologías ágiles para el desarrollo de software: Xtreme Programming (XP):  
<http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Alicante, U. (s.f). *Modelo vista controlador (MVC)*. Retrieved from Universidad Alicante:  
<http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Castillo Oswaldo, F. D. (05 de Febrero de 2016). *Programación Extrema*. Obtenido de  
<http://programacionextrema.tripod.com/fases.htm>
- Definicion.de. (19 de Enero de 2016). *Definicion.de*. Obtenido de <http://definicion.de/producto-terminado/>
- Design, I. (2016, Febrero 13). *PHP vs JAVA*. Retrieved from <https://itxdesign.com/php-vs-java/>
- Drupal, P. A. (2016, Febrero 21). *Por qué usar MySQL como tú gestor de Base de Datos*.  
Retrieved from <http://aprendiendodrupal.com/por-que-usar-mysql-como-tu-gestor-de-base-de-datos>
- Facilito, C. C. (2016, Febrero 24). *MVC (Model, View, Controller) explicado*. Retrieved from  
<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Geek, B. G. (2006, Marzo 23). *Software Libre y Software Propietario, ventajas y desventajas*.  
Retrieved from <http://www.gentegeek.com/sl-sp-ventajas-desventajas/>
- López, B. S. (2012). *COSTOS DE PRODUCCIÓN*. Retrieved from Ingenieria Industrial Online:  
<http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/producci%C3%B3n/costos-de-producci%C3%B3n/>
- Matanza, U. N. (2016, 01 28). *Método Agil XP*. Retrieved from Wiki Procesos de Software:  
<https://procesossoftware.wikispaces.com/>

Rodríguez, M. J. (2015, Julio). *Estudio Comparativo entre Metodologías*. Retrieved from

Universidad de Oviedo:

<http://digibuo.uniovi.es/dspace/bitstream/10651/32457/6/TFMMIJGarciaRodriguezRUO.pdf>

Romero, A. m.-F. (2016, Febrero 02). *Materiales o insumos de producción* . Retrieved from

Aula mass : <http://aula.mass.pe/manual/materiales-o-insumos-de-produccion>

Varo, E. b. (16 de 01 de 2015). *Una receta para calcular (bien) tus costos*. Obtenido de

<http://clientesmalos.com/2014/01/13/una-receta-para-calcular-bien-tus-costos/>

## **ANEXOS**

### **A. Estándares de desarrollo**

#### **Diseño de base de datos:**

1. Nombre de las tablas y campos deben ser descriptivos.
2. Para los nombres de las tablas y campos se debe utilizar el formato underscore para la separación de palabras.
3. Tablas deben estar normalizadas

#### **Arquitectura**

##### Back end

1. La aplicación está programada sobre Codeigniter framework PHP con arquitectura MVC:  
Se debe utilizar las funcionalidades, librerías, helpers que nos facilita el framework.
2. Los nombres de archivos para modelos, controladores, y vistas deben ser descriptivos de su función y en formato underscore si son nombres compuestos.
3. Los nombres de clases atributos y métodos deben ser descriptivos en su función y en formato underscore si son nombres compuestos.
4. Para el acceso a datos utilizar Active record que os proporciona el framework.

##### Front end

1. Utilizar JQuery o Java Script nativo para manipular datos del lado del cliente.
2. Utilizar la librería bootstrap que es el framework propuesto para la maquetación del sistema.
3. No se debe incluir código javascript en las vistas o controladores. El javascript o jquery debe estar dentro de un archivo js
4. Las librerías o archivos javascript, imágenes y css deben estar en la carpeta assets.

## B. Plan de entregas

Tabla 10

Detalle plan de entregas.

ITERACIÓN	NRO.	HISTORIA DE USUARIO	FECHA PLANIFICACIÓN		LANZAMIENTO	ESTADO DESARROLLO	PRUEBAS
			Inicio	Fin			
Primera	1	Registro de Usuarios	lun 11/01/2016	vie 22/01/2016	25/01/2016	Completo	Aprobado
Segunda	2	Registro de Productos	vie 29/01/2016	vie 05/02/2016	08/02/2016	Completo	Aprobado
	3	Registro de Insumos	mie 10/02/2016	mie 17/02/2016	22/02/2016	Completo	
	4	Recetas de Producción	mie 24/02/2016	mie 09/03/2016	14/03/2016	Completo	
Tercera	5	Creación de Órdenes de Producción	lun 04/04/2016	vie 22/04/2016	25/04/2016	Completo	Aprobado
	6	Amortización de Inversiones	mie 27/04/2016	jue 18/05/2016	20/05/2016	Completo	
	7	Planificación de Producción	lun 23/05/08	vier 10/06/2016	15/06/2016	Completo	