

**UNIVERSIDAD TECNOLÓGICA DE TIJUANA**

**TÍTULO DEL PROYECTO**

PRESENTA

MARÍA DE LOS ÁNGELES RABELERO CAMPOS

TIJUANA, B. C. ABRIL, 2014

TÍTULO A OTORGAR

LICENCIATURA EN INGENIERÍA EN

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

TRABAJO RECEPCIONAL

desarrollo para el control de inventarios en la industria restaurantera

**UNIVERSIDAD TECNOLÓGICA DE TIJUANA**

**TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN**



**TÍTULO A OTORGAR**

**LICENCIATURA EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN**

**TRABAJO RECEPCIONAL**

Realizada por

MARÍA DE LOS ÁNGELES RABELERO CAMPOS

En la empresa

OPERADORA DE FRANQUICIAS TAIKISHI S. A. DE C. V.

**Director de Trabajo Recepcional**

ING. ANTONIO REYES PÉREZ

Tijuana Baja California, 16 de abril de 2014

Dedicatoria y agradecimientos

Dedico este Trabajo Recepcional a mi asesor Ing. Antonio Reyes Pérez, a google, a mi fiel Kenzo y a mis padres por sus enseñanzas y consejos, que me han motivado para seguir mejorando mi ingenio y mi formación profesional.

Resumen

El presente Trabajo Recepcional propone el desarrollo de un software para el control de inventarios en la industria restaurantera, con el propósito de brindar a la operadora de restaurantes de comida Taikishi una aplicación empresarial que les auxiliara en el proceso de control de inventarios, facilitando la administración del abastecimiento de materiales y el cálculo de costos.

**abstract**

Índice

Página

[Dedicatoria y agradecimientos 3](#_Toc383244957)

[Resumen 4](#_Toc383244958)

[Índice 5](#_Toc383244959)

[Índice de figuras 7](#_Toc383244960)

[Índice de tablas 8](#_Toc383244961)

[Capítulo 1. Introducción 9](#_Toc383244962)

[1.1 Enunciado de la problemática 11](#_Toc383244963)

[1.2 Objetivos general y específicos 11](#_Toc383244964)

[1.3 Justificación 12](#_Toc383244965)

[1.4 Factibilidad 13](#_Toc383244966)

[1.5 Presentación de la metodología 15](#_Toc383244967)

[1.6 Plan de trabajo y cronograma de actividades 18](#_Toc383244968)

[Capítulo 2. Marco teórico 21](#_Toc383244969)

[2.1 Definición de inventarios 21](#_Toc383244970)

[2.2 Bases legales 29](#_Toc383244971)

[2.3 Aspecto técnico 30](#_Toc383244972)

[Capítulo 3. Marco contextual 40](#_Toc383244973)

[3.1 Antecedentes de la empresa 40](#_Toc383244974)

[3.2 Antecedentes informáticos 44](#_Toc383244975)

[Capítulo 4. Estrategia metodológica y resultados 46](#_Toc383244976)

[4.1 Fase de diseño 46](#_Toc383244977)

[4.2 Fase de desarrollo 57](#_Toc383244978)

[Capítulo 5. Conclusiones y recomendaciones 62](#_Toc383244979)

[5.1 Conclusiones 62](#_Toc383244980)

[5.2 Recomendaciones 66](#_Toc383244981)

[Referencias 68](#_Toc383244982)

[Anexos (opcional) 69](#_Toc383244983)

Índice de figuras

Página

[Figura 1. Modelo del ciclo de vida en cascada 16](#_Toc383244985)

[Figura 2. Modelo del ciclo de vida ágil 17](#_Toc383244986)

[Figura 3. Modelo del ciclo de vida híbrida-ágil 19](#_Toc383244987)

[Figura 4. Estructura de desglose del trabajo 21](#_Toc383244991)

[Figura 5. Clasificación de inventarios 22](#_Toc383244992)

[Figura 6. caracteristicas de las mercancias 23](#_Toc383244993)

[Figura 7. Procesos internos de los materiales 24](#_Toc383244994)

[Figura 8. Operaciones del inventario 28](#_Toc383244996)

[Figura 9. Arquitectura cliente – servidor orientada a web 32](#_Toc383244997)

[Figura 10. Patrón MVC 37](#_Toc383244999)

[Figura 11. Lista de comandos del CLI Artisan 38](#_Toc383245000)

[Figura 12. Diagrama de paquetes de los requerimientos 48](#_Toc383245001)

[Figura 13. Diagrama de paquetes de los requerimientos funcionales 48](#_Toc383245002)

[Figura 14. Modulo de compras 49](#_Toc383245003)

[Figura 15. Modulo de contabilidad 49](#_Toc383245004)

[Figura 16. Modulo empresa 50](#_Toc383245005)

[Figura 17. Modulo de Inventario. 50](#_Toc383245006)

[Figura 18. Modulo de mercancias 51](#_Toc383245007)

[Figura 19. Modulo de produccion 51](#_Toc383245008)

[Figura 20. Modulo de usuarios 52](#_Toc383245009)

[Figura 21. Modulo de ventas 52](#_Toc383245010)

[Figura 22. Diagrama de paquetes del dominio 53](#_Toc383245011)

[Figura 23. Modelo de dominio 53](#_Toc383245012)

[Figura 24. Administrador de ordenes 54](#_Toc383245013)

[Figura 25. Franquicia 54](#_Toc383245014)

[Figura 26. Cocina general 55](#_Toc383245015)

[Figura 27. Comisariato 55](#_Toc383245016)

[Figura 28. Gestionar empresa 56](#_Toc383245017)

[Figura 29. Gestionar mercancias 56](#_Toc383245018)

[Figura 30. Gestionar usuarios 56](#_Toc383245019)

[Figura 31. Diagrama E-R preliminar 57](#_Toc383245020)

[Figura 32. Actores de actividades 58](#_Toc383245021)

[Figura 33. Caracterisiticas del sistema 58](#_Toc383245022)

[Figura 34. Pila del producto 59](#_Toc383245023)

[Figura 35. Iteraciones por semana 60](#_Toc383245024)

[Figura 36. Asignacion de actividades 60](#_Toc383245025)

Índice de tablas

Página

[Tabla 1. Lista de precios de Soft Restaurant® 16](#_Toc383245026)

[Tabla 2. Programa de Hitos 21](#_Toc383245030)

[Tabla 3. Asignación de Roles 21](#_Toc383245031)

[Tabla 4. Calendario de Actividades 21](#_Toc383245032)

[Tabla 5. Características de la clasificación de mercancías 26](#_Toc383245037)

[Tabla 6. Conectores UML 34](#_Toc383245040)

# Introducción

A través del tiempo las empresas y negocios han hecho uso de herramientas para auxiliarles a llevar acabo sus tareas administrativas y de contabilidad. Con los avances tecnológicos estas herramientas se han convertido en programas y computadoras que han revolucionado la manera de llevar acabo la administración en las empresas y son de vital importancia para su funcionamiento. Controlan sus procesos y recursos, automatizan tareas cotidianas, aumentan la productividad y disminuyen costos de producción.

Actualmente en el mercado hay muchas herramientas que tratan de dar solución a todas las necesidades de administración de las empresas. Cada empresa tiene necesidades únicas y por lo tanto procesos únicos. El querer adaptar el flujo de trabajo de alguna de estas herramientas existentes para satisfacer los requerimientos de algún proceso único de alguna empresa, puede convertirse en una labor difícil de realizar, ya que se requiere conocer la tecnología usada en crear la herramienta y se debe contar con el licenciamiento adecuado para realizar las modificaciones.

En ocasiones es más factible para una empresa el desarrollar su propia herramienta hecha a las medidas de sus necesidades y que pueda adaptarse a los cambios del negocio con facilidad. Para diseñar e implementar una herramienta se requieren aplicar las tecnologías de información. Las tecnologías de la información se refieren a los programas, computadoras y otros medios que sirven para almacenar y procesar datos y documentos de la empresa de manera automatizada, eficiente y segura.

El entender las tecnologías de la información existentes permite seleccionar los recursos tecnológicos y procedimientos más apropiados para solucionar un problema, en este caso el desarrollo de una herramienta que realicé ciertas funciones, bajo condiciones específicas que serán explicadas con detalle en los futuros apartados.

Una de las funciones que toda herramienta debe poseer es el manejo de inventarios, ya que la compra y venta de bienes y servicios es una de las principales actividades en toda empresa. En razón a lo anterior, el presente trabajo recepcional describe el desarrollo de una herramienta de control de inventarios, que permita registrar materiales sencillamente, hacer más fácil la elaboración de reportes y mantener un control más estrictico sobre los materiales.

En las siguientes secciones del trabajo recepcional se darán a conocer los conceptos teóricos y tecnológicos para comprender la funcionalidad y utilidad de la herramienta. Este se organiza en 5 partes que serán explicadas a continuación.

El capítulo I explica brevemente como se origina el problema, se plantea una solución, su alcance, sus objetivos, se define la metodología que se seguirá y el plan de trabajo.

El capítulo II primero explica brevemente los conceptos de Inventarios y su sistema de control, luego las bases legales de los inventarios en las empresas mexicanas, y por último las partes técnicas que integran la solución del problema.

El capítulo III da una reseña de la empresa y describe la situación actual del área en la empresa donde se originó el problema y donde se desarrolla la aplicación.

El capítulo IV documenta el desarrollo de la solución propuesta, sus funcionalidades y los resultados obtenidos que serán comparados con los objetivos previamente establecidos.

El capítulo V presenta las conclusiones del trabajo y se proponen mejoras futuras.

## Enunciado de la problemática

La Operadora de Franquicias Taikishi S. A. de C. V. cuenta con 4 franquicias de restaurantes y una cocina general que prepara salsas y complementos de platillos para las franquicias. En estos establecimientos existen fallas en el registro y control de mercancías. En el proceso actual las franquicias envían una lista de mercancías y complementos de platillos a suplir en una de cálculo (Excel) mediante correo electrónico a la bodega principal “el Comisariato”. Los materiales son entregados a cada franquicia, pero al no llevarse un control interno en la bodega y en sus franquicias se ocasionan problemas de hurto y desperdicio de materiales.

## Objetivos general y específicos

Introducción.

### Objetivo general

Analizar las necesidades de la empresa para diseñar e implementar una aplicación empresarial web para el control de inventarios, con la finalidad de tener acceso a la información en tiempo real. Utilizando los recursos tecnológicos disponibles con el propósito de minimizar el tiempo de implementación y el costo de inversión.

### Objetivos específicos

1. Elaborar el análisis de requerimientos y la selección de tecnología más adecuada para el desarrollo de la aplicación empresarial web.
2. Seleccionar una arquitectura que le permita ser flexible y escalable.
3. Modelar la base de datos conforme a los requerimientos de almacenamiento y manipulación de datos.
4. Diseñar vistas amigables e intuitivas, que faciliten al usuario el interactuar con el sistema.
5. Definir el esquema de seguridad para acceder a la aplicación empresarial web, y que garantice la disponibilidad del mismo.

## Justificación

En la actualidad, la empresa Operadora de Franquicias Taikishi S. A. de C.V., desconoce las cantidades de mercancías que se van usando en los procesos de preparación de platillos, en la cocina general y sus franquicias. Una nueva administración ha enfatizado el impacto negativo que produce la falta de control en los inventarios y cómo afecta seriamente la rentabilidad y ganancias de la empresa.

Por tal motivo busca implementar una solución adecuada para llevar un estricto control en las entradas y salidas de mercancías. Implementando una aplicación empresarial web para el control de inventarios se tendrá la información de los materiales que se encuentran en la bodega principal y los establecimientos. Lo que le permitirá a la empresa conocer las áreas de oportunidades para mejorar sus procesos operativos y controlar pérdidas y desperdicios.

Al implementar la aplicación empresarial web se pretende obtener los siguientes beneficios:

* Llevar a cabo consultas de los inventarios en base a sus características, ubicación, estado y responsable.
* Auxiliar en la auditoria de las mercancías en los establecimientos.
* Emitir reportes en base a los registros de órdenes de compra y las entregas de material.
* Apoyar al área de administrativa en procesos de control del personal.
* Brindar la oportunidad de agregar módulos para la administración de la empresa en un futuro.

En el desarrollo de la aplicación empresarial web se verán reflejados los conocimientos adquiridos en mi educación universitaria como el análisis de requerimientos, el modelamiento de base de datos, desarrollo web y administración de proyectos informáticos.

## Factibilidad

Un proyecto es factible cuando se sabe que es posible desarrollarlo con los recursos disponibles, la factibilidad del éxito se conoce analizando los siguientes aspectos [Cruz, Granados, Lizama, Rivas, 2011]:

* Factibilidad Técnica.
* Factibilidad Económica.

### Factibilidad técnica

En la factibilidad técnica se evalúan el hardware que servirá de servidor para la solución y el software que se usara para desarrollar la solución propuesta.

En la evaluación del hardware se dio a conocer que la empresa no cuenta con un servidor. El contar con un servidor es fundamental para poder acceder a la aplicación. Las computadoras que se usan en los establecimientos son de mediana capacidad, con procesador Pentium, poca memoria, y con el sistema operativo XP, lo cual no las hace un candidato para usarse como servidor para la aplicación. Una solución sería el comprar un servidor, otra solución sería el rentar un servidor compartido entre varias personas a una compañía de hospedaje web.

En la evaluación del software se vio que la falta de un servidor no permitía tener la información disponible entre los establecimientos, la información de cada establecimiento estaba aislada. Al emplear una solución orientada a la web la información de los establecimientos estará centralizada en un hospedaje web. Se usaran plataformas de desarrollo de licenciamiento libre, que automatizarían muchas tareas del desarrollo.

En conclusión, al usar los servicios de un hospedaje web y plataformas de desarrollo con licenciamiento libre, el proyecto es factible técnicamente ya se entregara una solución cumpliendo con los objetivos establecidos, con un costo mínimo para la empresa y en el tiempo fijado.

### Factibilidad económica

Para determinar si el proyecto es factible económicamente se hará una comparación entre el costo de extender el licenciamiento del sistema existente para controlar los inventarios y el costo de desarrollar el proyecto.

**Extender el licenciamiento del sistema existente**

Actualmente la empresa paga una renta mensualmente de un sistema llamado Soft Restaurant® que lleva el control y administración del restaurante, incluyendo el inventario. Solo permite un usuario a la vez y guarda la información de manera local. Para dar solución a la problemática y poder compartir la información entre los establecimientos se tendrían que cambiar el tipo de licenciamiento del programa., lo cual requiere de una fuerte inversión inicial. La tabla 1 muestra el costo total del nuevo licenciamiento.

Tabla 1.

Lista de precios de Soft Restaurant®

|  |  |  |
| --- | --- | --- |
| Licencia | Cantidad | PRECIO |
| Soft Restaurant® 8.0 UPGRADE a versión Enterprise desde STANDARD (Licencia Módulo Administrativo Central + sistema Soft Restaurant) Solo aplica para la licencia principal, no incluye licencias de las sucursales que deben adquirirse adicionalmente. | 1 | $ 30,000.00 |
| Soft Restaurant® 8.0 UPGRADE 1 Lic. Sucursal a versión Enterprise desde STANDARD | 2 | $ 7,500.00 |
| COSTO TOTAL | | $ 45,000.00 |

Fuente: http://www.css-mex.com/Archivos/Lista de precios SR Cliente.pdf

**Aplicación empresarial web**

Para la implementación de la aplicación no habrá necesidad de inversiones adicionales a solicitar los servicios de hospedaje web, el cual tiene un costo de 14 dólares mensuales. Al ser una aplicación web, se utilizara la infraestructura tecnología existe sin preocuparse por las características de las computadoras, ya que la aplicación corre sobre cualquier dispositivo.

En conclusión al comparar ambas propuestas se poder ver que la alternativa de la aplicación empresarial web es la más viable económicamente.

## Presentación de la metodología

Es importante elegir un modelo de ciclo de vida de desarrollo de software adecuado ya que todas las actividades del proyecto se derivan de este proceso. Este modelo define las fases por las que pasa un proyecto de desarrollo de software. La metodología que se seguirá para desarrollar la aplicación se basa en una combinación del modelo de cascada y del modelo ágil.

En la metodología de cascada las etapas son secuenciales y lineales. Una etapa no empieza hasta que la otra termine. Este es el modelo más antiguo y sirve como base para otros modelos de ciclos de vida. La figura 1 muestra las fases de este modelo.

Figura 1.

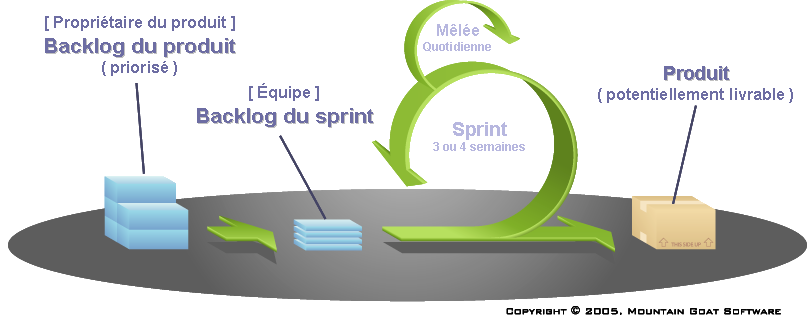
Modelo del ciclo de vida de la metodología de cascada

En una metodología ágil las etapas son cíclicas y duran de 1 a 4 semanas, permitiendo un mejor control del desarrollo. El proyecto comienza con propuestas de tareas del proyecto, las tareas aceptadas se colocan en la pila del producto o “Backlog del producto”. Al comienzo de cada etapa llamada iteración o “sprint” el equipo decide que tareas de la pila del producto incorporar en la Pila de la iteración o “Backlog del sprint”.

Cada día de la iteración el equipo se reparte las tareas que puede cumplir ese día. En cada iteración o “sprint” se cumplen una funcionalidad del proyecto, hasta terminar con un producto completo. La figura 2 muestra las fases de este modelo.

Figura 2.

Modelo del ciclo de vida de la metodología ágil



Fuente: http://fr.wikipedia.org/wiki/Scrum\_(méthode)

Esta combinación dará como resultado una metodología híbrida-ágil.

Se usara la metodología de cascada para las fases de la planificación, análisis de requerimientos y diseño. Y se usara la metodología ágil llamada Scrum para las fases de desarrollo del código, pruebas y lanzamiento. A continuación se detalla cada una de las fases de la metodología híbrida-ágil adaptada.

**Planificación**

En esta fase se identifica el patrocinador del proyecto, las prioridades del proyecto, el alcance, los objetivos y los recursos con que se cuentan. Se hace el estudio la factibilidad para llevarlo a cabo y se definen los roles de los participantes en el proyecto.

**Análisis de requerimientos**

En esta fase se entrevistan los usuarios finales, se fijan los objetivos que la empresa requiere del sistema y se detallan sus funcionalidades. Se debe asegurar de que los que solicitan el programa comprendan claramente el alcance del proyecto.

**Diseño**

En esta fase se modela la aplicación en base a las necesidades de la empresa, se toman decisiones acerca de la arquitectura del proyecto, el patrón de diseño que se seguirá y en que plataforma se usara para programarse. También se diseñan los mecanismos de seguridad del sistema y sus reglas de acceso.

**Desarrollo**

En esta fase se inicia el proceso de codificación y depuración del código. Se desarrolla la interfaz de usuario y la lógica de la aplicación. En esta fase se deciden que metodologías aplicar para desarrollar el código.

**Pruebas**

En esta fase se simula el código mediante pruebas unitarias, se obtiene una idea de la rapidez del mismo y de la facilidad o dificultad que tendría el agregar un nuevo módulo al sistema, así como seria su mantenimiento. También se podrá hacer la observación de algún cambio por parte del cliente a la aplicación.

**Lanzamiento**

En esta fase el lanzamiento del código es incremental a cada sprint de la metodología ágil. En cada incremento se realiza la documentación y los respaldos. Una vez que se realizan el último sprint el sistema está listo para ser liberado, se instala, en este caso un hospedaje web, se forma a los usuarios y se cierra el proyecto.

A continuación en la siguiente figura, se muestra el diagrama del ciclo de vida de la metodología híbrida-ágil.

Figura 3.

Modelo del ciclo de vida de la metodología híbrida-ágil

## Plan de trabajo y cronograma de actividades

A continuación se muestra varias tablas que detallan las actividades en el plan de trabajo, y las fechas programadas para cada etapa del proyecto.

La tabla 2, muestra la terminación de las tareas más significativas del proyecto. Cada hito representa de un entregable del proyecto.

Tabla 2.

Programa de Hitos

|  |  |  |
| --- | --- | --- |
| No. | Hito | Fecha Programada |
| 1 | Culminación de la planificación | 17/01/14 |
| 2 | Culminación del análisis | 24/01/14 |
| 3 | Culminación del diseño del sistema | 14/02/14 |
| 4 | Culminación del desarrollo | 28/03/14 |
| 5 | Culminación del proyecto | 04/04/14 |

La tabla 3, presenta la asignación de responsabilidades y tareas de cada rol y la forma como se repartirá el trabajo entre los integrantes.

Tabla 3.

Asignación de Roles

|  |  |
| --- | --- |
| RECURSO HUMANO | ROL |
| Adán Prudencio | Analista, DBA |
| María Rabelero | Programador, Tester |

La tabla 4 presenta una lista de todas las actividades del proyecto con sus fechas previstas de comienzo y final.

Tabla 4.

Calendario de Actividades

|  |  |  |  |
| --- | --- | --- | --- |
| FECHA | ACTIVIDAD | Rol | HORAS |
| 13/01/14 – 17/01/14 | Planificación | Analista | 40 |
| 20/01/14 – 24/01/14 | Análisis de Requerimientos | Analista | 40 |
| 27/01/14 – 14/02/14 | Diseño de Arquitectura | Programador | 120 |
| 17/02/14 – 28/03/14 | Desarrollo de Código | Programador | 240 |
| 31/03/14 – 04/04/14 | Cierre del Proyecto | Equipo | 40 |
| Total de horas | | | 480 |

La figura 4 presenta la estructura de desglose del trabajo (WBS). El WBS es una descomposición jerárquica orientada al entregable del trabajo a ser ejecutado en paquetes por el equipo de proyecto.

Figura 4.

Estructura de desglose del trabajo

# Marco teórico

Este capítulo se divide en tres secciones. La primera sección presenta los conceptos administrativos para llevar el control de inventarios. La segunda sección presenta las regulaciones gubernamentales que rigen los inventarios en las empresas mexicanas. Y la tercera sección explica la parte técnica del sistema.

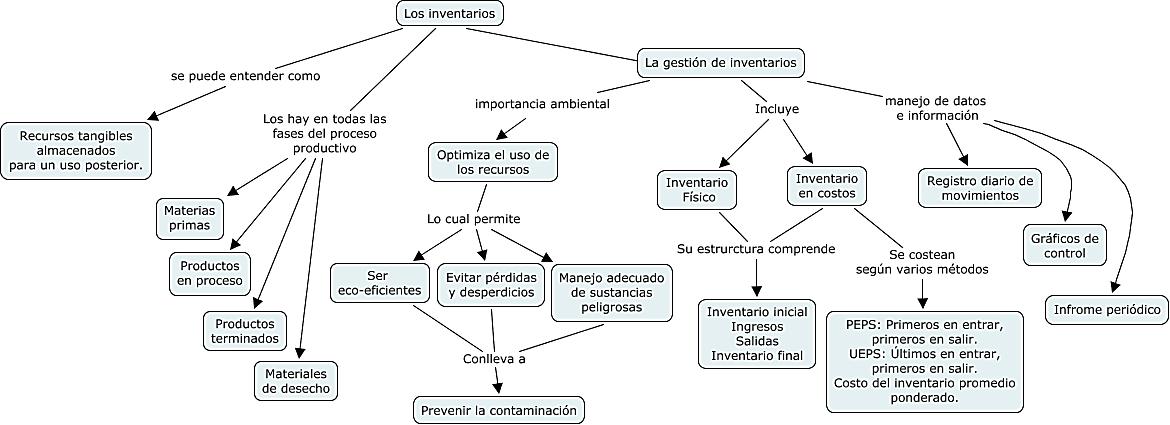
## Definición de inventarios

Un inventario es una relación detallada de bienes tangibles o mercancías que una empresa mantiene almacenados para su venta o para ser usados en la producción de otros bienes o servicios [Rumbea, 2003].

Los inventarios de acuerdo a las características físicas de los objetos a contar se pueden clasificar en diferentes tipos. La figura 5 muestra un mapa mental de las maneras como se pueden clasificar los inventarios.

Figura 5.

Clasificación de inventarios



Fuente: http://goo.gl/XKmYgf

Para la elaboración de un sistema para el control de los inventarios se requiere de una clara compresión del comportamiento de los elementos que conforman un inventario.

* Una **mercancía** es un objeto material cuya transformación, venta o producción es el objeto principal de la empresa. Estas se clasifican en artículos y productos.
* Un **artículo** es una mercancía con la que empresa comercia. No es para el consumo propio, sino es destinado directamente a la venta.
* Un **producto** es un objeto hecho en la empresa para el consumo propio o para su venta. Un producto consta de uno o varios artículos.
* Un **stock** es artículo en existencia o disponible en un almacén.
* Un **inventario** es el total de stocks en un almacén.

### Mercancías

Aunque el control de inventarios es el objetivo del sistema, el inventario no existe como tal, sino es el resultado de un control sobre las mercancías en un almacén. Las características de las mercancías deben estar bien definidas. La siguiente figura muestra las características de una mercancía.

Figura 6. Características de las mercancías

Fuente: Gestión de Inventarios. Universidad Simón Bolívar. 2008.

**Mercancía para procesar**

Es una mercancía que es comprada para ser usada en los procesos internos de la empresa y no para su venta directa. La figura 7 muestra los momentos dentro de los procesos internos donde se generan inventarios.

* **Inventario de materias primas.** Las mercancías han sido adquiridos pero no han entrado en el proceso de fabricación.
* **Inventario de producto en proceso**. Están en procesamiento, estarán en esta fase mientras son procesados y transportados para su venta. En esta fase se producirán mermas.
* **Inventario de productos terminados.** Listos para la venta.

Figura 7.

Procesos internos de los materiales

**Fuente: Gestión de Inventarios. Universidad Simón Bolívar. 2008.**

Es importante mantener una clasificación del porcentaje de merma que una mercancía puede producir. La clasificación es:

1. Totalmente merma
2. 5-30%
3. 1-5%
4. Sin influencia

* Mercancía de Valor

El método ABC permite clasificar las mercancías que son compradas para procesar y para su venta directa, de acuerdo a su importancia en la empresa [Pareto, 1897].

* **Artículo clase A.** Son los artículos más importantes. Son pocos, su valor es muy alto, y requieren de más control.
* **Artículo clase B.** Son artículos importantes. Los volúmenes de ventas son considerables, y se controla automáticamente.
* **Artículo clase C.** Son artículos poco importantes. Su volumen es bastante alto, de poco valor, y no requieren de controles sofisticados.

La tabla 5 exhibe con detalle las características de las mercancías y como deben administrarse dependiendo de su clasificación.

Tabla 5.

Características de la clasificación de mercancías

|  |  |  |
| --- | --- | --- |
| TIPO DE CLASE | POLÍTICAS DE CONTROL | MÉTODOS DE CONTROL |
| Clase A  • Los más importantes  • Realmente pocos | • Cubrimiento de existencias por semana.  • Aproximación a Justo a Tiempo.  • No exceder los 300 artículos. | • Monitoreo constante y control estricto.  • Registros de información precisos.  • Políticas de control. |
| Clase B  • Importantes | • Control clásico.  • Cubrimiento de existencias entre 2 y 8 semanas. | • Monitoreo básico.  • Registros por excepciones. |
| Clase C  • Poco importantes | • Supervisión mínima y control simple.  • Cubrimiento de existencias entre 3 y 20 semanas.  • Pedidos bajo orden. | • Monitoreo simple.  • Gestión para evitar agotados y eliminar excesos. |

Fuente: José Fernández. Mejoramiento Del Control De Inventarios. Universidad Autónoma De Occidente. Santiago De Cali. 2008.

* **Mercancías por características**

La unidad básica de almacenamiento o stock keeping units (SKU) es una forma de identificación única de las mercancías precisando las características primarias y particulares de cada artículo en relación a las generales.

* Material que lo constituye.
* Dimensiones.
* Presentación o empaque.
* Referencia: marca, nombre del fabricante.
* Unidad: kilogramo, litro, centenar, millar, etc.
* Los utilizadores de la mercancía.
* Clasificación o Familia.

### Almacén

El objetivo de un almacén es permitir a la empresa, tener disponible y de una manera constante los recursos necesarios para llevar a cabo sus actividades normalmente y actuar con eficacia. Sus funciones principales son:

* Control del stock
* Distribución de mercadería o producto terminado
* Uso de sistemas para una mejor gestión de almacenes

Los almacenes deben cubrir las necesidades del ciclo productivo y deberán almacenar materias primas, en producción y terminadas. También deberán satisfacer las necesidades del ciclo de distribución y almacenamiento de los artículos y productos. Los almacenes se clasifican como:

* Almacenes principales o centrales.
* Almacenes subsidiarios o periféricos.
* Almacenes de producción.

### Costos de los inventarios

Para implementar un sistema de inventario se deben considerar aquellos costos que varían directamente relacionados con la operación de la empresa. Básicamente existen cinco tipos de costos, los cuales son:

**Costo de pedido o aprovisionamiento (k)**

Es el costo de poner una orden al proveedor.

* De adquisición. Valor de la compra y transporte.
* De reaprovisionamiento. Costo de pedidos, inspecciones de calidad.
* De fabricación. Ordenes, mermas de arranque.

**Costo de inventario (h)**

Es el costo de mantener los artículos inventariados.

* Por riesgo. Seguros, perdidas, mermas, deterioros, robos, etc.
* Del local. Alquiler, vigilancia, suministros, servicios, etc.
* De transporte Interno instalaciones, maquinarias, etc.
* De conservar en buen estado los productos. Frío, humedad, etc.

**Costo por déficit (d)**

Generado por falta de productos en el inventario cuando la venta se pierde.

* Riesgo de perder una venta (merma de contribución).
* Disconformidad insatisfacción del cliente.
* Trabajo de procesar y despachar una orden parcial.

**Costos asociados a la capacidad (c)**

Son asociadas con el personal: contrataciones, entrenamiento, despidos.

**Costo total (t)**

Es la suma de los costos mencionados anteriormente.

C (t) = C (k) + C (h) + C (d) + C (c)

### Operaciones en el inventario

La eficiencia del proceso de un sistema de inventarios es el resultado de la buena coordinación entre las diferentes áreas de la empresa. La figura 8 muestra las operaciones que involucra las transacciones dentro del inventario.

Figura 8.

Operaciones del inventario

### Gestión de inventarios

La gestión de Inventarios es todo lo relativo a técnicas, métodos e instrumentos orientados al control y manejo de inventarios. Sirve para evaluar los procedimientos de registros, compras, entradas y salidas. [Fernández, 2008].

En la mayoría de los casos la demanda es independiente y el tiempo de suministro es conocido pero no constante y se corra el riesgo de una ruptura de stock, ya que los valores reales fluctuarán. Para disminuir el riesgo se lleva un stock de seguridad (SS) que absorberá las fluctuaciones.

* **Nivel de servicio** (NS) el stock de seguridad dependerá del nivel de servicio que se desee. A mayor nivel mayor tamaño del stock, menor posibilidad de que se produzcan roturas y mayor capital inmovilizado.
* **Riesgo de ruptura** (RR) es el complementario del nivel de servicio. La probabilidad de que la demanda sea superior a la prevista.
* **Punto de pedido** (PP) es un pedido que se lanza. Se calcula:

PP = (*Demanda de una mercancía* (diaria) X *Tiempo de suministro*)   
+ *Stock de seguridad*

### niveles óptimos de existencias

Los pasos para la planeación de los niveles óptimos de existencias de materiales se determinan por medio de un plan logístico que determina cuánto y cuándo reabastecer los materiales [Narasenhan, 1996]. Estos pasos son:

**Determinar las existencias.**

Son los procesos que informan sobre las existencias físicas de los artículos. Algunos procesos son:

* Toma física de inventarios, auditoria de existencias.
* Evaluación de los procedimientos de recepción y ventas.

**Análisis de inventarios.**

Se harán los siguientes análisis estadísticos para medir el beneficio que producen las existencias.

* Sistema de clasificación ABC.

**Control de costos.**

Obtener los datos necesarios para calcular el costo del inventario.

**Políticas de punto de reabastecimiento.**

Usando un sistema de gestión de inventarios emplear fórmulas del sistema probabilísticos para:

* Fijar políticas de punto de pedido (PP).
* Establecer políticas de seguridad y prever riesgos de faltantes.

## Bases legales

Este Proyecto se basa legalmente en leyes vigentes de los Estados Unidos Mexicanos del año 2014.

La Ley del impuesto sobre la renta (ISR) establece las personas morales tienen obligaciones relativas a los controles de inventarios. En el Capítulo VII (De las obligaciones de las personas morales) artículo 71, fracción XIV.

Llevar un control de inventarios de mercancías, materias primas, productos en proceso y productos terminados, según se trate, conforme al sistema de inventarios perpetuos...

El Código fiscal de la federación (CFF) establece que las personas que están obligadas a llevar contabilidad, deben llevar un inventario. En el Titulo Segundo (De los derechos y obligaciones de los contribuyentes) Capitulo Único, artículo 28, fracción I.

La contabilidad, para efectos fiscales, se integra por los libros, sistemas y registros contables, papeles de trabajo, estados de cuenta, cuentas especiales, libros y registros sociales, control de inventarios y método de valuación…

* 1. También estable que son infracciones relacionadas con la obligación de llevar contabilidad el no llevar un inventario. En el Titulo Cuarto, Capítulo I (De las infracciones) artículo 83, fracción II.

No llevar algún libro o registro especial a que obliguen las leyes fiscales; no cumplir con las obligaciones sobre valuación de inventarios o no llevar el procedimiento de control de los mismos, que establezcan las disposiciones fiscales.

## Aspecto técnico

Esta sección empezara explicando que es un sistema informático. Luego se explicaran los lenguajes usados en el diseño (UML) y la programación (PHP) de la aplicación. Las librerías de programación y por último los programas usados.

### Sistemas Informáticos

Un sistema informático está compuesto por hardware, software y usuarios que procesan e interpretan la información. Posee una arquitectura cliente – servidor que estructura el sistema en dos partes: la información del sistema se guarda en el servidor, el software que se ejecuta en el servidor responde a las peticiones de información de varios clientes, la comunicación entre las dos partes es mediante protocolos de comunicación estandarizados [Berson 1992].

Una aplicación empresarial es un sistema informático crucial para el funcionamiento de un negocio. Se utilizan para hacer las tareas más importantes del negocio. Tienen las siguientes características [Anderson, 2012]:

* Capacidad de leer y almacenar datos centralizadamente.
* Un flujo de trabajo estructurado.
* Interfaz de usuario.
* Capacidad para aplicar las reglas del negocio.
* Validación de datos.
* Capacidad de hacer reportes.
* Autenticación y autorización de usuarios.

Una aplicación empresarial web es un sistema informático ejecutado en el entorno web, donde el cliente y el servidor se encuentran separados físicamente. En el entorno web se usan servidores web, de base de datos, de archivos y de aplicaciones, los clientes son navegadores web o aplicaciones que usen protocolos que maneje un servidor web y los protocolos de comunicación son HTTP, FTP, SMPT y SSL, como se muestran en la en la siguiente figura.

Figura 9.

Arquitectura cliente – servidor orientada a web

Base de Datos

**Aplicación**

Los servidores usados en esta aplicación son Apache y PostgreSQL. Apache es un servidor web que se encarga de enviar páginas web en la World Wide Web usando el protocolo HTTP. Su arquitectura es modular, se puede configurar fácilmente para usar diversos módulos que aportan funcionalidades.

* El módulo mod\_php permite a Apache interpretar archivos PHP para generar y servir páginas web dinámicamente.
* El módulo mod\_dbd ofrece conexiones de base de datos a los módulos que requieran conectarse a alguna base de datos SQL.

PostgreSQL es un sistema de gestión de bases de datos que permiten el almacenamiento, modificación y extracción de la información de su base de datos relacional. Usa un lenguaje de consultas llamado PL/PgSQL. Es bien conocido por sus características nativas:

* Concurrencia (permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos).
* Tipos (Texto de largo ilimitado, Figuras geométricas, Direcciones MAC e IP, Arreglo, Moneda, Fechas con usos de horarios.).
* Varios esquemas de tablas en un dominio, aserciones (assertions) y transacciones, entre otras.

### Los Lenguajes Usados

**Lenguaje de diseño**

El lenguaje unificado de modelado (UML) es un estándar que define un lenguaje visual usado para modelar sistemas informáticos mediante diagramas y entidades. Los modelos se construyen a partir de elementos. En cada etapa del diseño se van usando distintos elementos. La tabla 6 describe los elementos usados en el diseño de la aplicación.

Tabla 6.

Elementos UML

|  |  |
| --- | --- |
| SÍMBOLO | DESCRIPCIÓN |
|  | Un paquete es un nombre de espacio. Solamente los elementos en los paquetes pueden ser miembros del nombre de espacio. |
|  | Una entidad es un objeto o concepto con existencia independiente, es decir, se diferencia unívocamente de otro objeto o cosa. |
|  | Un elemento enumeración es un tipo de dato, cuyas instancias pueden ser cualquiera de un número de literales de enumeración definidas por el usuario |
|  | Una interfaz garantiza soportar un comportamiento entre clases que usan la interfaz, lo cual permite al sistema tratar elementos no relacionados de la misma manera, a través de una interfaz común. |
|  | Un elemento estructurado representa un arreglo de actividades relacionadas. Esta vinculado a un diagrama hijo con actividades que tienen relaciones. |
|  | Un requisito es algo que se requiere que el sistema tenga, o algo que la aplicación haga. |

Fuente: http://www.sparxsystems.com.ar/download/ayuda/index.html?umlconnections.htm

Las relaciones entre elementos se representan mediante conectores. La tabla 7, describe los conectores usados en el diseño de la aplicación.

Tabla 7. Conectores UML

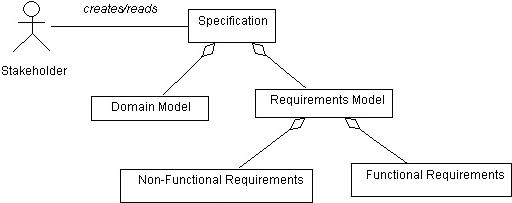
|  |  |
| --- | --- |
| SÍMBOLO | DESCRIPCIÓN |
|  | Una relación de agregación es un tipo de asociación que muestra que un elemento contiene o está compuesto de otros elementos. |
|  | Una generalización se usa para indicar herencia. El origen hereda las características del destino. |
|  | Una asociación entre dos elementos implica una asociación como una variable en una clase. |
|  | Una asociación dirigida muestra propiedad (tiene, un avión *tiene* pasajeros), la asociación solo es navegable en un sentido. |

Fuente: http://www.sparxsystems.com.ar/download/ayuda/index.html?umlconnections.htm

La figura 10 muestra el proceso de diseño. De las especificaciones brindadas por el cliente del proyecto se crea el modelo de dominio y el modelo de requerimientos. Los requerimientos se dividen en funcionales y no funcionales. Los funcionales representan la parte de programación, los no funcionales representan la parte física necesaria para que el sistema pueda ser ejecutado.

Figura 10.

Proceso de diseño UML



Fuente: http://www.cs.sjsu.edu/~pearce/uml/SoftwareEngineering.htm

A partir del modelo de requerimientos funcionales se modelan dos grupos de diagramas.

* Los diagramas estructurales que muestran una vista estática del modelo. Por ejemplo el diagrama de clases, el diagrama de objetos, el diagrama de paquetes, el diagrama de estructura compuesta.
* Los diagramas de comportamiento que muestran una vista dinámica del modelo. Por ejemplo el diagrama de actividades y el diagrama de casos de uso.

A continuación se describen los diagramas usados en el diseño de la aplicación.

El diagrama de Requisitos Funcionales describe las características del sistema, al vincularlos a otros componentes del sistema se demuestra cómo el requisito es cumplido estableciendo una relación entre las pruebas y los requisitos.

El modelo del Dominio define objetos físicos y abstractos de un área específica del sistema. Se puede usar para visualizar las relaciones entre ellos y las responsabilidades de las clases conceptuales (estas clases representan un grupo de cosas, no son clases que definen objetos de programación).

El diagrama de Estructura Compuesta se usa para expresar arquitecturas en tiempo de ejecución, patrones de uso, y las relaciones de los elementos participantes, los que pueden no estar reflejados por diagramas estáticos.

**Lenguaje de Programación**

En las aplicaciones con arquitectura cliente – servidor se usa un leguaje del lado del servidor y otro del lado del cliente. En la aplicación web las páginas web que se servirán a los clientes usaran los lenguajes HTML, JavaScript y CSS. Los procesos que definen la lógica del servicio en el servidor usaran el lenguaje PHP.

**HTML**

El Lenguaje de Marcas de Hipertexto (HTML) es un estándar que define un lenguaje de notación. El lenguaje usa notaciones o etiquetas que definen elementos de tipo imagen, enlace (a otras páginas), formulario, párrafo, lista, encabezado, bloque, etc. Cuando un navegador analiza un documento HTML crea objetos de sus elementos, formando una página web desplegable o audible.

Las páginas web formadas solo de elementos HTML son estáticas, solo despliegan contenido, como un periódico o una revista, no se puede interactuar con una página estática. Los elementos de las etiquetas HTML están programadas en el navegador, no todos los navegadores estas programados igual, así que no todos entienden las etiquetas de la misma y los despliegan de diferentes formas.

La última versión HTML 5 incorpora nuevas etiquetas para el uso de videos, geo-localización, almacenamiento local, marcado de teléfonos, envió de emails y mensajes SMS, validación de formularios, entre otros. Siendo una nueva versión tomara tiempo para que se incorporen todas estas etiquetas a todos los navegadores.

**CSS**

Hojas de Estilo en Cascada (CSS) es un lenguaje de hojas de estilo para controlar la presentación visual de un documento escrito en un lenguaje de marcas, como HTML. Mediante reglas se dan valores de color, tamaño, posición, visibilidad, efectos, tipografía y lectura a los elementos de una página web. El documento CSS es interpretado por el navegador, pero no todos los navegadores están programados igual y cada uno brinda una presentación diferente.

Se les llaman en cascada por la manera como determinan qué valor se aplica a un elemento, ya que un documento CSS puede definir varias reglas para un elemento basado en una de tres posibilidades, por orden de precedencia estos son: el atributo id, el atributo de clase, el nombre del elemento.

CSS 2 permite presentar un documento con distintos formatos para diferentes tipos de dispositivos, tamaño de pantalla, tipo de pantalla o impresoras. La última versión CSS 3 añade la capacidad de animar y transformar elementos, usar gradientes, sombras, opacidad, redondear esquinas entre otras.

**JavaScript**

Es un lenguaje de script dinámico usado para manipular objetos y estilos de una página web. Al igual que HTML, también es interpretado por el navegador. Este lenguaje tiene sintaxis, patrones de diseño, es orientado a objetos, pero no usa clases, sino prototipos.

Ajax usa JavaScript para comunicarse con el servidor web y desplegar el resultado en un instante ya que no requiere recarga toda la página, solo el objeto donde se muestra el resultado. Este resultado usualmente es enviado del servidor en la notación de objetos JavaScript (JSON), un formato ligero para el intercambio de datos.

Con la popularidad de Ajax se desarrollaron librerías e infraestructuras (frameworks) para JavaScript que simplifican mucho su programación y le añaden muchas funcionalidades.

**PHP**

El Pre-procesador de Hipertexto (PHP) es un lenguaje de script, esto implica que los scripts son interpretados y no compilados. Cuando el cliente hace una petición al servidor web pidiendo una página, el servidor ve si hay código PHP, si lo hay ejecuta el intérprete de PHP. Éste procesa las tareas del código, generando una respuesta. La respuesta es regresada al servidor, quien a su vez se la envía al cliente.

La respuesta generada puede variar. Por ejemplo pudiera ser código HTML o JSON para desplegar información de una base de datos, o código JavaScript para redirigir al cliente a otro sitio web, o código PHP para incluir archivos que muestran una platilla de página de error, o pudiera un archivo.

PHP fue creado por una persona para su página web personal y no como un lenguaje de programación su acrónimo era Personal Home Pages. PHP fue desarrollándose hasta llegar a PHP 5, la última versión estable, que ofrece un soporte para la programación orientada a objetos, como el objeto controlador PHP (PDO) para conexión a bases de datos.

### Librerías de programación

Se usaron muchas librerías en la elaboración de la aplicación. Esta sección explicara el funcionamiento de las librerías más importantes de PHP y JavaScript.

**Librerías PHP**

* **Laravel**

Laravel no es una librería PHP, sino una infraestructura (framework) de aplicaciones PHP. Laravel automatiza las tareas habituales de los sistemas informáticos web como la autenticación, enrutamiento, sesiones y el almacenamiento caché. Imitando la arquitectura y el comportamiento de otros frameworks web como Ruby on Rails, ASP.NET MVC y Sinatra.

Proporciona potentes herramientas: un patrón MVC, enrutamiento RESTful, un ORM llamado Eloquent, un contenedor de control inverso, un sistema de migración de base de datos, pruebas unitarias, un motor de plantillas llamadas Blade, una CLI llamada Artisan, y numerosos paquetes de diversas funcionalidades.

* + **MVC**

Este patrón de arquitectura para software agrupa los datos y lógica, la interfaz gráfica, y el módulo de comunicación en tres componentes: el modelo, la vista y el controlador (MVC), como es mostrado en la figura.

1. El modelo representa la lógica de la aplicación. Escucha y responde las peticiones del controlador de leer y almacenar en la base de datos información de los objetos del sistema.
2. El controlador pide cambios al modelo, al recibir la respuesta la pasa a la vista. Un controlador sabe a qué vista llamar dependiendo del objeto modelo que este interactuando.
3. La vista recibe peticiones con información del controlador y se encarga de desplegarla usando una plantilla. Algunas vistas son creadas dinámicamente a partir de los objetos del modelo.

Figura 11.

Patrón MVC

**Petición del cliente**   
GET, POST, PUT, DELETE  
(sobre HTTP)

**Respuesta al cliente**   
HTML, JSON  
(sobre HTTP)

**1. Petición al modelo**

**2. Respuesta del modelo**

**3. Petición a la vista**

**Modelo**  
ORM, clases

**Vista**  
plantillas

**Controlador**enrutamiento

* **Rutas RESTful**

Un localizador de recursos uniforme (URL) se usa para localizar recursos en Internet, es como la dirección del recurso. La transferencia de estado representacional (REST) es un diseño orientada a recursos en el que las páginas web son vistas como servicios web y pueden ser identificados por sus direcciones URL.

Laravel permite definir una ruta para manejar todas las acciones relacionadas a un controlador usando las convenciones REST. Por ejemplo, al crear un controlador para gestionar los recursos "fotos", se crean varias rutas URL para ejecutar las acciones REST del recurso “fotos”. También se crean métodos para cada una de estas acciones, con notas informándole cuales URL y verbos manejan. La tabla 8 muestra los recursos generados del controlador “fotos”.

Tabla 8.

Acciones del controlador de recursos “fotos”

|  |  |  |  |
| --- | --- | --- | --- |
| HTTP | URL | ACCIÓN | Uso |
| GET | /fotos | index | Despliega todas las imágenes |
| GET | /fotos/create | create | Regresa una forma para crear una foto |
| POST | /fotos | store | Crea una foto |
| GET | /fotos/1 | show | Despliega la foto “1” |
| GET | /fotos/1/edit | edit | Regresa una forma para editar la foto “1” |
| PUT | /fotos/1 | update | Modifica la foto “1” |
| DELETE | /fotos/1 | destroy | Borra la foto “1” |

Fuente: http://laravel.com/docs/controllers#restful-controllers

* **ORM**

Es una herramienta que realiza un mapeo objeto-relacional (ORM), es decir, refleja la información de los objetos de las clases y la base de datos relacional, creando la sensación de que se está trabajando con una base de datos de objetos. Se ahorra el tener que escribir código para crear, leer, modificar y eliminar (CRUD) información de la base de datos de todas las clases, ya que la herramienta se encarga de crear los métodos y actualizarlos.

* **Contenedor IoC**

El Contenedor de Control Inverso (IoC) es una herramienta para manejar las dependencias de las clases. En el patrón de Inyección de Dependencias (DI) al iniciar la aplicación se crea un contenedor con instancias de todas las clases y las suministran o “inyecta” a las clases cuando las requieran, para que estas no tengan que crear los objetos. Con IoC una dependencia es creada e inyectada al tiempo de ejecución de la clase que la requiera, lo que permite flexibilidad en dependencia que pueden variar y cambiar.

* **Migraciones**

Las migraciones se usan para llevar el control sobre los cambios en la base de datos mediante código PHP. Esta herramienta permite modificar el esquema de base de datos o regresar el esquema a un estado anterior. Las migraciones se suelen combinar con el Constructor de esquemas para gestionar fácilmente el esquema de la aplicación. Como agregar o eliminar un campo en una tabla, o tablas enteras o un dominio de tablas.

* **Pruebas unitarias**

Una prueba unitaria es una prueba que se hace a un pedazo de código por separado del resto del sistema. Esto sirve para asegurar que cada uno de los módulos funcione correctamente. Laravel incorpora PHPUnit encargado de generar las pruebas y establecerá automáticamente el entorno de configuración para pruebas, proporcionando sesiones falsas y otra información que no afectara a la aplicación.

* **Plantillas**

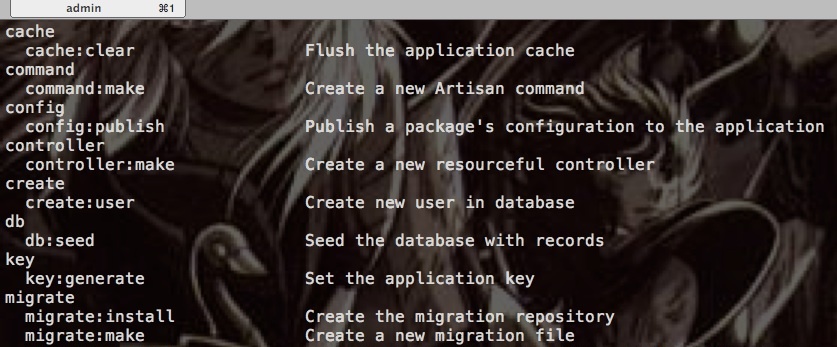
Blade es un motor de plantilla diseñado para combinar una o más plantillas con un modelo de datos para producir un documento HTML final. Las plantillas mezclan HTML con una sintaxis especial que permite usar variables, funciones, bucles y condiciones para desplegar información más fácilmente.

* **CLI**

Una Interfaz de Línea de Comandos (CLI) es un medio para enviar comandos a un programa mediante una línea de comandos. Artisan proporciona una serie de comandos de ayuda para el desarrollo de código. La figura 12 muestra algunos de los comandos disponibles en el CLI.

Figura 12.

Lista de comandos del CLI Artisan



* 1. **Composer**

Composer no es una librería, sino una herramienta para la gestión de librerías en proyectos PHP. Las librerías se declaran en un archivo, al ejecutar Composer este descarga e instala cada librería en un directorio dentro el proyecto, también crea un archivo que carga las librería con la posibilidad de usar un nombre de espacio. Se puede especificar una versión de la librería, o verificar si hay actualizaciones para descárgalas, instalarlas y cargarlas al proyecto automáticamente.

**Librerías de JavaScript**

* **jQuery**

JQuery ofrece una serie de funcionalidades basadas en JavaScript que si se codificaran para cada proyecto requerirían de mucho código. En otras palabras simplifica mucho la programación y reduce el tiempo de desarrollo. Actualmente hay miles de librerías que usan las funciones de jQuery para lograr resultados en poco tiempo y hacerlas más ligeras al compartir las mismas funciones.

* **AngularJS**

Es un framework de JavaScript de permite extender el vocabulario HTML con directivas y atributos propios como: ng-app, ng-controller, ng-model, ng-view. Permite hacer consultas Ajax con peticiones HTTP, tiene su motor de plantillas, y permite manipulación de datos en JSON.

* **Modernizr**

Realiza pruebas para detectar las características de HTML 5 y CSS 3 no soportadas por el navegador, crea un objeto llamado "Modernizr" que contiene los resultados booleanos de las pruebas, y crea clases con los nombres de las características no soportadas. Este objeto y las clases pueden ser usados para agregar elementos o librerías que proporcionan las características que no están integrados en el navegador (polyfills).Su

* **Less**

Less (Leaner CSS) es un lenguaje de hoja de estilos dinámico diseñado por Alexis Sellier. Less es un pre-procesador de CSS que extiende el lenguaje CSS, es decir, al procesar el archivo con código less se obtiene otro archivo con código CSS. Less permite usar variables, anidamiento, operadores, mixins y funciones.

* **Twitter Bootstrap**

Es una librería basada en jQuery contiene código HTML, Less, tipografía, formularios, botones, navegación y muchos componentes gráficos. Cuenta con varias funciones jQuery propias, como un calendario y un carrusel. Se puede adaptar el archivo de Bootstrap, seleccionando los componentes que deseen usar en el proyecto y aplicar un estilo a todos ellos.

### Herramientas de PROGRAMACIÓN

En esta sección se describirán los programas que se usaron para modelar, desarrollar, gestionar el desarrollo y el control de versiones del código del sistema.

* **Enterprise Architect de Sparx Systems**

Es una herramienta de ingeniería de software asistido por computadora (CASE) usado en el diseño y construcción de sistemas de software, y en el modelado de procesos de negocios. Está basada en la especificación UML 2.1. En este programa se diseñaron los requisitos organizados como un modelo jerárquico. Y se diseñó el modulo central mediante el modelo del dominio.

* **Sublime Text**

Es un editor de texto y editor de código fuente. Aunque no es un entorno de desarrollo integrado (IDE) su funcionalidad es extensible con paquetes construido por la comunidad. Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto. En este programa se escribió el código del sistema.

* **IceScrum**

Es una herramienta para la gestión de un proyecto mediante Scrum. Permite agregar actores, historias a la cartera del proyecto, dividir el tiempo en iteraciones y añadir actividades a las historias de cada iteración. Este programa se usó para programar los objetivos y las tareas diarias de codificación.

* **Git**

Git es un sistema de control de versiones distribuido diseñado para manejar proyectos con rapidez y eficiencia. Git permite sincronizar, clonar repositorios, cometer con comentarios cada cambio, y compartir el código en GitHub. GitHub es un servicio de hosting web para los proyectos de desarrollo de software que utilizan el Git para el control de versiones. En este sitio está el repositorio donde se guarda cada cambio en el código que se va generando.

# Marco contextual

En este capítulo se analizan las condiciones existentes en la empresa. Se hará un estudio en estas instalaciones debido a que se quieren observar los procedimientos que se siguen, con la finalidad de saber a detalle las actividades que abarcan el almacenaje de mercancías.

Se comienza con una descripción de la empresa y luego se describirá el resultado del estudio de las instalaciones donde se ubica geográficamente el problema.

Ejemplo:

**Pobreza en Guatemala**

Marco contextual:

* Estudios elaborados sobre pobreza en Guatemala.
* Situación de pobreza en el país.

## antecedentes

Comprende indicar algunos de los autores que han investigado el tema, qué métodos o técnicas utilizaron y qué resultados obtuvieron.

## caracteristicas espacio - temporales

Contextualizar un trabajo de investigación es describir dónde (lugar o ambiente) se ubica el fenómeno o problema de investigación.

Elaborar una descripción del entorno o ambiente en que se produce o desarrolla un fenómeno.

De esto que deje abajo se puede rescatar y estructurar información.

La compañía nombre es una empresa local constituida en 2007 como restaurante y operador de franquicias. Cuenta con un nombre bien conocido en la región, como un restaurante de 4 estrellas y con un gran número de clientes nacionales e internacionales. En los restaurantes se ofrece tanto comida japonesa como tailandesa. La decoración ostenta un moderno estilo japonés. Su concepto de competitividad se centra en valores, eficiencia y calidad en todos los productos y servicios.

Durante los últimos años dicha empresa se ha empeñado en optimizar sus operaciones, para lo cual ha implementado sistemas de información que le permitan manejar la información del negocio en forma sistemática.

La información de cada restaurante fue obtenida mediante una pequeña entrevista a la persona encargada del almacén principal y una visita a un restaurante, a la cocina general y al almacén principal. Los datos que se preguntaron fueron: nombre, descripción, dirección y teléfono del restaurante, nombre del capitán de meseros, número de meseros que laboran, número de turnos y la forma en que se reparte la propina.

La investigación se llevó a cabo en los almacenes de los restaurantes. Los almacenes no son muy grandes, cuentan con estantes de varias repisas para poner los productos, la altura de estos estantes no es muy grande y la distribución de dichos estantes es muy buena y está bien definida como debe estar cualquier almacén. Los productos se encuentran divididos según el grupo al que pertenecen y por fecha para que la ubicación de los mismos sea los más pronto posible.

El resultado de este estudio fue un muy detallado conocimiento de los requerimientos y necesidades que la aplicación debe cumplir.

### Diagnóstico de la situación actual

En la necesidad de la empresa por disminuir el costo del inventario y optimizar la cantidad de producto disponible para las ventas, se requiere como primera medida disminuir el costo del inventario de lento movimiento mediante acciones e identificación del mismo. Se encontró con la situación crítica de la empresa, en la manera de hacer sus pedidos y la falta de control en las cantidades de productos en inventario.

La empresa actualmente realiza sus pedidos de forma empírica, sin tener en cuenta la demanda real ni las tendencias de sus productos, basándose en promedios de venta simplemente. Por esta razón, la intención del proyecto es implementar un sistema, el cual, teniendo en cuenta factores relevantes de demanda y tiempo de pedidos, pronostique y sugiera la cantidad adecuada de producto a comprar para satisfacer la demanda del mercado.

Una vez suministrado el análisis de demanda y el sistema para pronóstico de los ítems seleccionados para el proyecto, se pretende optimizar los pedidos, tener control sobre las cantidades en inventario y tener una herramienta que le permita a la persona encargada de realizar los pedidos en la compañía, basarse en la misma para no incurrir en costo o perder negocios por faltantes en el inventario.

# Estrategia metodológica y resultados

En este capítulo se detallan el proceso de la elaboración del sistema. El capítulo está divido en dos partes: Fase de diseño y Fase de desarrollo.

* La primera parte comprende el diseño en alto nivel de la arquitectura. En esta etapa se desarrollaron los diagramas de requerimientos, el diagrama de dominio y el diagrama preliminar de la base de datos.
* La segunda parte comprende el desarrollo de la aplicación basada en el modelo de la arquitectura desarrollada en la fase de diseño.

## Fase de diseño

Después de analizar la problemática y definir la solución se realiza el proceso de modelamiento. El objetivo en esta fase es asimilar los requerimientos esperados de la solución y plasmarlos en la definición y especificación de los diagramas UML, los cuales nos ayudaran a comprender cómo es que trabajara e interactuara el sistema con el usuario final.

### Modelado de requisitos

* **El Modelo de Requerimientos**

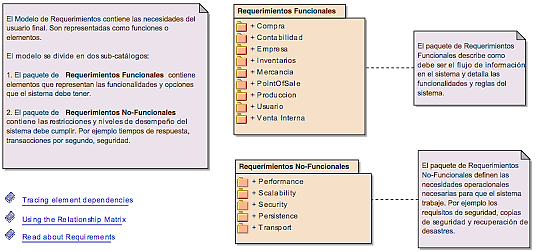
Contiene las necesidades del usuario final. Son representadas como funciones o elementos. El modelo se divide en dos sub-catálogos:

1. El paquete de requerimientos funcionales contiene elementos que representan las funcionalidades y opciones que el sistema debe tener.
2. El paquete de requerimientos no-funcionales contiene las restricciones y niveles de desempeño del sistema debe cumplir. Por ejemplo tiempos de respuesta, transacciones por segundo, seguridad.

En la siguiente figura se muestra el diagrama de paquetes de los requerimientos.

Figura 13.

Diagrama de paquetes de los requerimientos

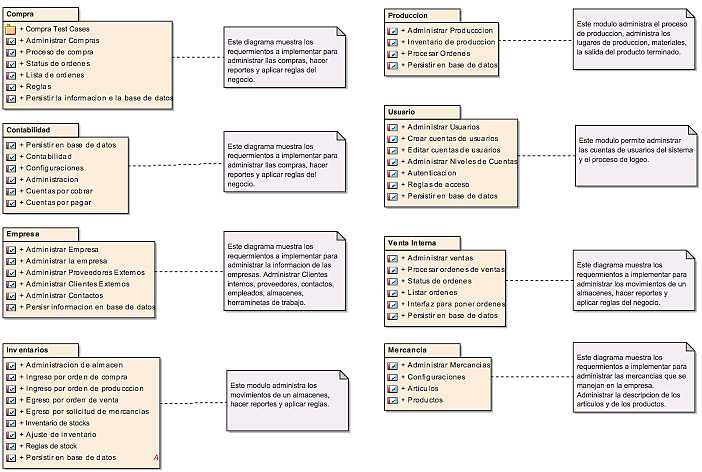


* **Requerimientos funcionales**

Los requerimientos funcionales del sistema describen lo que se espera que el sistema sea capaz de hacer. El modelo describe las tareas que tiene hacer, procesos que administrar, las reglas que sigue y características.

Figura 14.

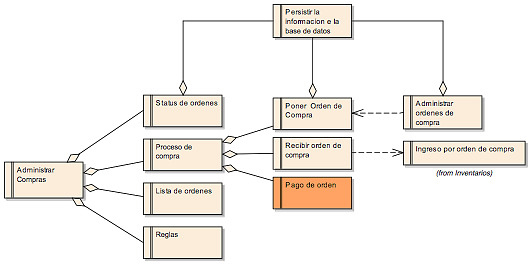
Diagrama de paquetes de los requerimientos funcionales



### Modelado de los módulos

* **Módulo de compras.** Se espera que el modulo se capaz de administrar las órdenes de compra, control de facturas y pendientes por recibir.

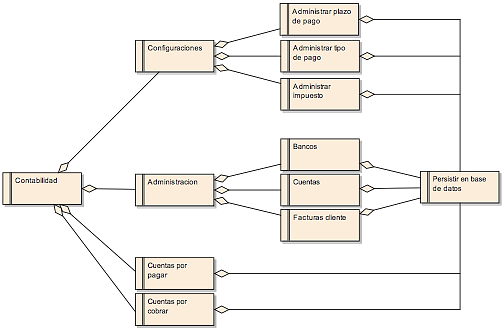
Figura 15. Módulo de compras



* **Módulo de contabilidad.** se espera que el sistema se capaz de administrar las cuentas, cuentas por pagar, administrar facturas, impuesto.

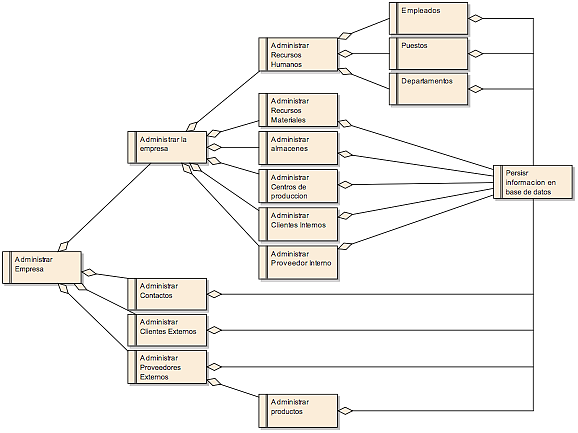
Figura 16.

Módulo de contabilidad



* **Modulo empresa.** se espera que el sistema se capaz de administrar clientes internos, proveedores, contactos, empleados y almacenes.

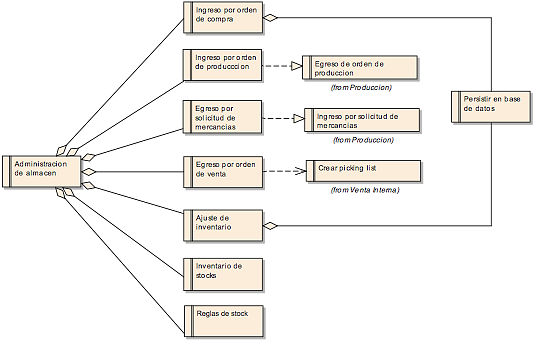
Figura 17. Modulo empresa



* **Módulo de Inventario.** se espera que el sistema se capaz de administrar los movimientos de un almacenes, reportes y reglas del negocio.

Figura 18.

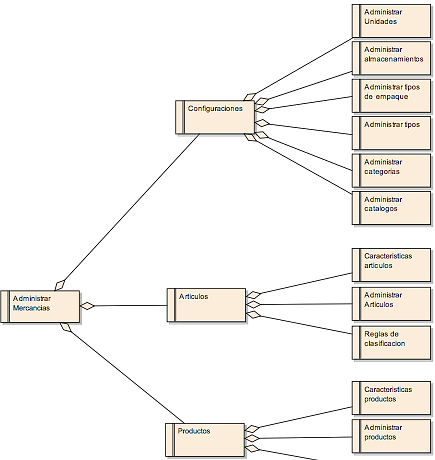
Módulo de Inventario.



* **Módulo de mercancías.** se espera que el sistema se capaz de administrar mercancías, precios, categorías y catálogos.

Figura 19.

Módulo de mercancías



* **Módulo de producción.** se espera que el sistema se capaz de llevar el proceso de producción, materiales, la producto terminado.

Figura 20.

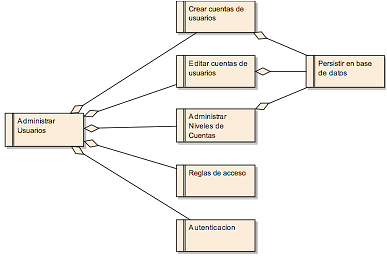
Módulo de producción



* **Módulo de usuarios.** se espera que el modulo se capaz de administrar acl, niveles, roles, cuentas de usuarios del sistema y el proceso de autenticación.

Figura 21.

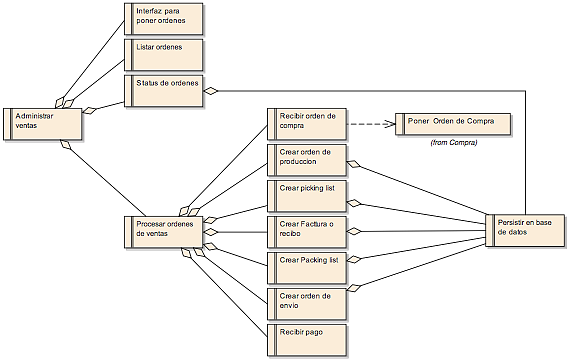
Módulo de usuarios



* **Módulo de ventas.** se espera que el modulo se capaz de administrar la tienda, mercancías por enviar, status de una orden, administrar envíos

Figura 22.

Módulo de ventas

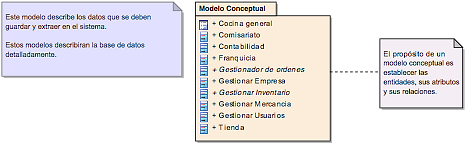


### Modelado del sistema

El dominio del sistema muestra las entidades principales del negocio. Este análisis favorecerá la arquitectura final y será necesaria para su construcción.

Figura 23.

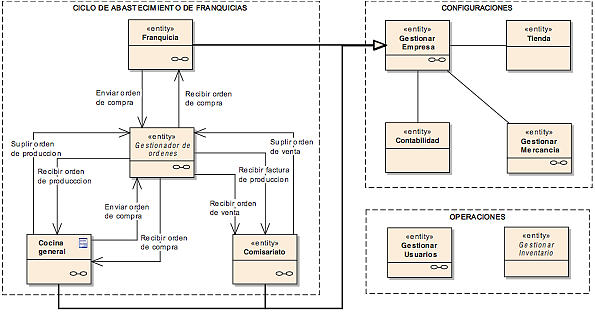
Diagrama de paquetes del dominio



El modelo del dominio se puede utilizar para documentar lazos entre y responsabilidades de clases conceptuales. Las clases conceptuales representan el concepto de un grupo de entidades.

Figura 24.

Modelo de dominio

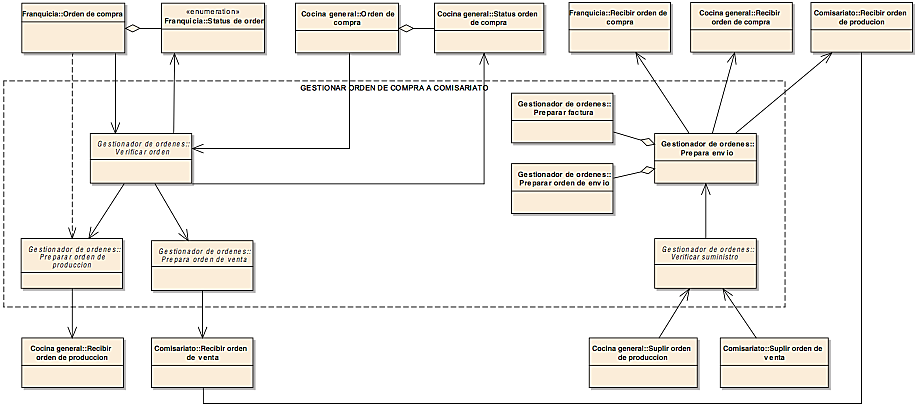


En el modelo de dominio cada entidad es un diagrama compuesto, compuesto por una relación de otras entidades. Se mostrara cada diagrama que compone una entidad y se explicara el objetivo.

* **Administrador de órdenes.** es el modulo central de la operación, administra las compras, reportes y aplicar reglas del negocio y movimientos del almacenes. Como lo muestra la figura 25.

Figura 25.

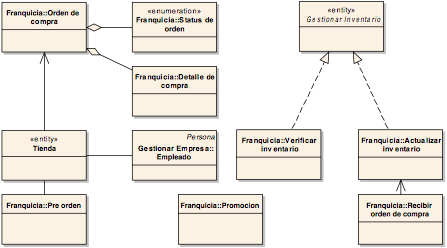
Administrador de órdenes



* **Franquicias**. El modelo conceptual de los clientes internos muestra las actividades de los empleados, almacenes y herramientas de trabajo.

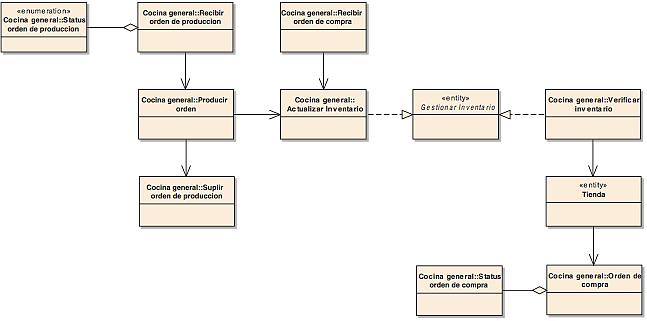
Figura 26.

Franquicia



* **Cocina general.** El modelo conceptual de la muestra la implementación de actividades para administrar el proceso de producción, administra los lugares de producción, materiales, la salida del producto terminado.

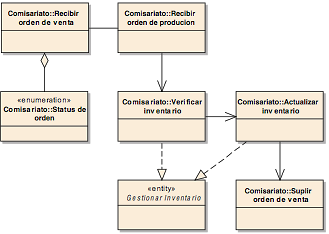
Figura 27. Cocina general



* **Comisariato**. El modelo conceptual del almacén general se muestra como se suplen las órdenes de compra de las franquicias.

Figura 28.

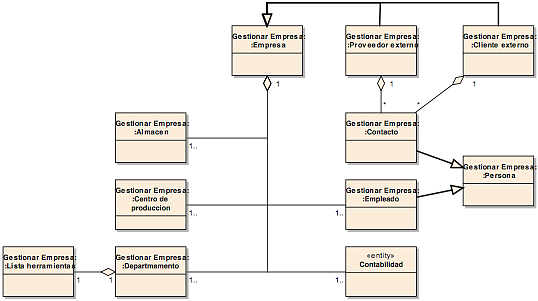
Comisariato



* **Gestionar empresa.** El diagrama muestra los requerimientos a implementar para administrar la información de las empresas. administrar clientes internos, proveedores, contactos, empleados, almacenes, herramientas de trabajo.

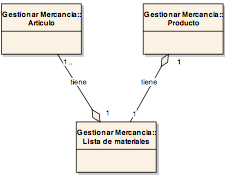
Figura 29.

Gestionar empresa



* **Gestionar mercancías**. El diagrama muestra la implementación de actividades para administrar las mercancías que se manejan en la empresa. Administrar la descripción de los artículos y de los productos.

Figura 30. Gestionar mercancías



* **Gestionar usuarios**. El diagrama muestra la implementación de actividades para administrar las cuentas de usuarios del sistema.

Figura 31. Gestionar usuarios

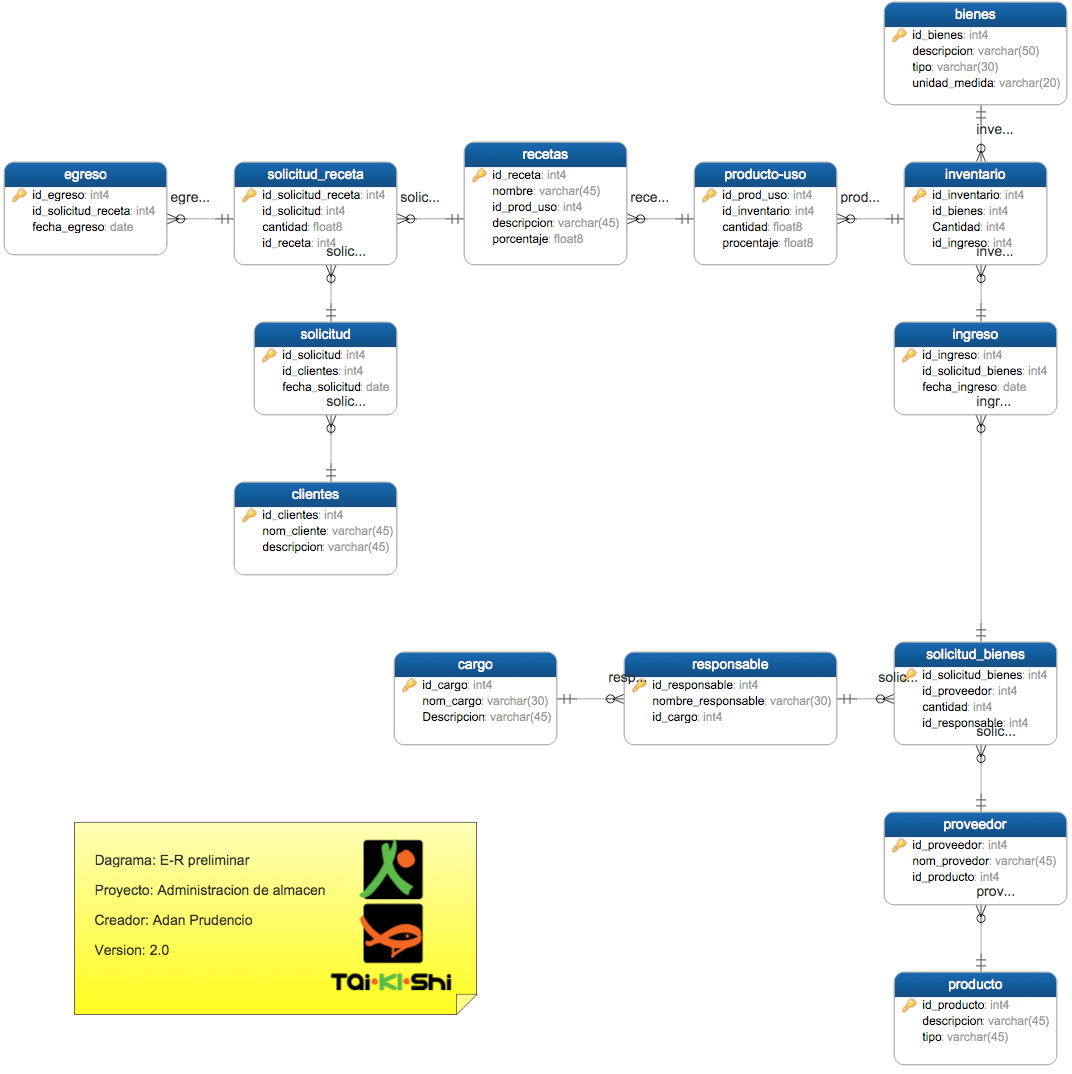


### Modelado preliminar de la base de datos

Se realizó un modelo preliminar con las relaciones entre las tablas que compondrán el sistema. Conforme se vayan desarrollando las iteraciones se harán cambios al modelo y a la base de datos. La figura 32, muestra el primer diseño entidad – relación de la base de datos.

Figura 32.

Diagrama E-R preliminar



## 

## Fase de desarrollo

El objetivo principal de la fase es desarrollar la aplicación que se diseñó anteriormente, consiste con los requerimientos fundamentales para satisfacer las necesidades del cliente.

### Planificar las iteraciones

Esta sección es importante ya que las iteraciones deben producir un producto completo. Se describirán los pasos de la planificación de las iteraciones.

1. **Se crearon los actores.**

Se comenzó con los actores para poder usarlos al crear las historias.

Figura 33.

Actores de actividades



1. **Se crearon las características.**

El crear las características del sistema no es un paso necesario, pero sirve para agrupar las historias.

Figura 34.

Características del sistema



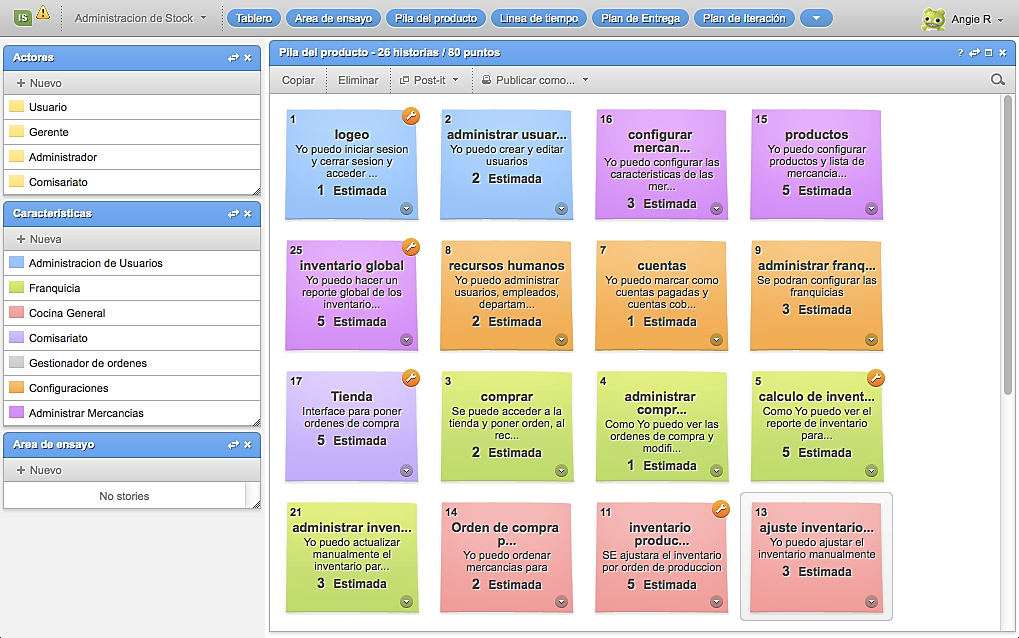
1. **Se creó la pila del producto.**

Se puede observar que el programa da la opción de marcar las historias como servicios o como actividad que un actor puede cumplir. También permite poner un número a la historia para distinguir la dificultad y el tiempo estimado que tomara desarrollar la actividad.

La figura 34 muestra todas las historias que se tienen que realizar para construir el sistema, después de haber sido catalogadas por característica y acomodada por importancia y urgencia.

Figura 35.

Pila del producto



1. **Se crearon las iteraciones.**

Una vez que se tiene lista la pila del proyecto. El programa da la opción de auto asignarlas a las iteraciones, basándose en la capacidad de producción del equipo de trabajo. El resultado son iteraciones donde el promedio estimado de las historias es similar.

La figura 36 muestra las historias planeadas en iteraciones por semana.

Figura 36.

Iteraciones por semana



1. **Se asignaron actividades.**

Una vez que se tienen la iteración, se puede acceder a una iteración y se pueden asignan actividades a cada historia. En cuanto se empiece a trabajar en una actividad, se moverá a la columna en progreso. Una vez completada se coloca en la columna de terminada.

La figura 37, muestra unas actividades en espera y las actividades recurrentes, como las pruebas unitarias que se harán en cada actividad.

Figura 37.

Asignación de actividades

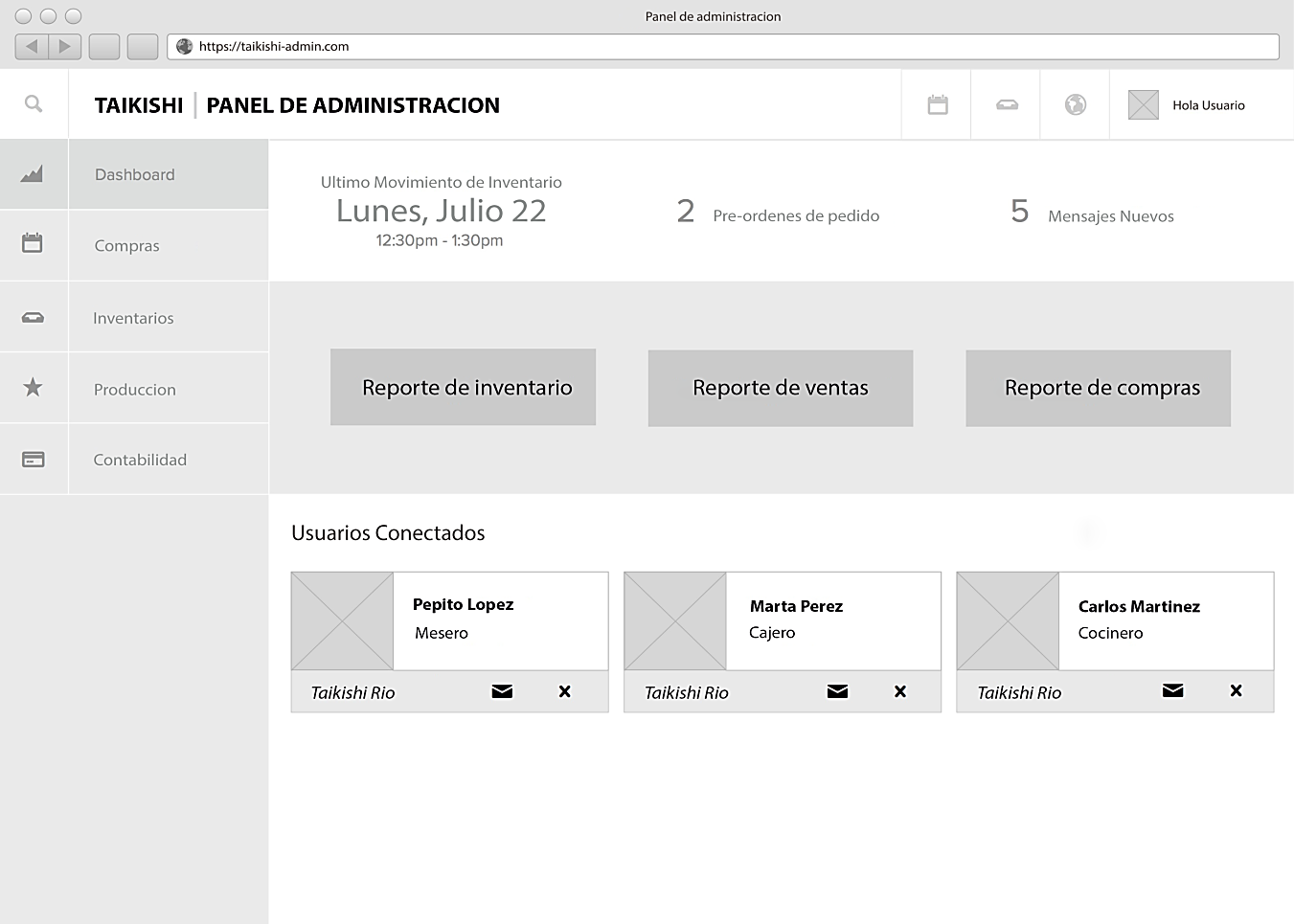


### Diseño de la interfaz gráfica

En esta sección se exponen los criterios para el diseño del diseño gráfico Web. El objetivo de este paso es crear un estándar de interfaz gráfica para las vistas del sistema y para uniformizar el aspecto visual y la interacción con el usuario. Se seguirá el patrón gráfico mostrado en la figura 38.

Figura 38.

Patrón de diseño gráfico del sistema



### Desarrollo del programa

En esta sección se realizó la construcción de software y comprende las labores de codificación y pruebas del producto a partir de las pautas definidas en los la fase de diseño.

### Cierre del proyecto

Esta sección tiene como propósito la puesta del sistema en producción (afinando las pruebas integrales) y la capacitación de los usuarios. A su vez se completará la documentación final del sistema.

* + **Pruebas de integración**

En la fase de pruebas, se relacionó toda la información que solicitó el cliente contra el resultado de software obtenido. Una vez ensamblados todos los módulos del sistema, se verificó que todo funcionara correctamente cumpliendo con los requerimientos establecidos.

* + **Documentación y presentación**.

# Conclusiones y recomendaciones

Este capítulo final comprende las conclusiones y recomendaciones una vez completadas las fases del proyecto. Lo propuesto en esta trabajo recepcional es un sistema de inventarios que presente la información de inventarios reciente y controle la totalidad de los artículos para la venta y producción.

La implementación del sistema permitirá compartir datos entre los departamentos y franquicias de la empresa. Para mantener un nivel de servicio de abastecimiento óptimo, y reducción del costo del inventario. Por lo tanto la implementación de un sistema de información agiliza procesos y disminuye tiempos de espera.

## Conclusiones

Este proyecto fue concebido con el objetivo de desarrollar un sistema para la gestión de inventarios. Con él se consiguió implementar una solución capaz de administrar los inventarios, almacenes, la producción, ventas internas y externas entre las franquicias y el almacén general.

* + Los esfuerzos y el tiempo invertido en el análisis y diseño de la solución posibilitaron la cobertura de todos los requerimientos funcionales. Logrando cumplir con las funcionalidades deseadas.
  + La incorporación de buenas prácticas y de la metodología Scrum en la fase de desarrollo permitieron cumplir con los tiempos de entrega en cada una de las iteraciones.
  + Este proyecto comprueba la capacidad de integración de aplicaciones construidas bajo PHP y PostgreSQL. logrando una significativa reducción de costos en la solución y cumpliendo los requerimientos no funcionales en cuanto la arquitectura.
  + La adopción de Laravel como framework de desarrollo y Twitter Bootstrap permitió una mejor implementación de funcionalidades desde una interfaz gráfica intuitiva, orientada a eventos y provista de una serie de controles Web.
  + Se puede concluir que el sistema será de gran ayuda para la implementación de la organización. La aplicación realizará funciones que son de suma importancia para llevar el correcto control y funcionamiento de la organización, en la mayoría de los casos muchas empresas realizan estos cálculos manualmente haciendo estos procesos muy tediosos y con un alto grado de equivocación en el cálculo de las operaciones.

## Recomendaciones

* Se recomienda el uso del sistema integral propuesto. El personal que lo use debe responsabilizarse de incluir en su totalidad los artículos que la empresa use y los mantenga al día.
* El desarrollo del sistema integral concluye con la elaboración de tablas de base de datos relacional, aplicable a lenguajes con esta plataforma. Por lo tanto, recomiendo se elaboren programas que exploten en su totalidad la información, para que a su vez la toma de decisiones por la alta dirección de la empresa en base a esta información, sea oportuna y acertada.
* Finalmente, se recomienda establecer un esquema de seguridad en la base de datos, mediante la creación de grupos personalizados de usuarios. Labor que debe ser diseñada en conjunto con el administrador de base de datos de la compañía para asegurar el acceso fiable a la base de datos.

Referencias

Anexos (opcional)

Los anexos se pueden respetar su presentación original.