Proyecto Ekflugo

Sistema POS para el área de Retail

Contenido

# Introducción

El presente trabajo tiene como objetivo sentar las bases para el desarrollo de un sistema de Punto de Venta (POS) para el área de retail en general. Lo que se persigue con este sistema es entrar al mercado de productos informáticos para el comercio minorista en general, con una solución que cumpla con estos tres requisitos: de fácil uso, expandible y económico.

El nombre clave del proyecto es Ekflugo, que significa “despegue” en idioma esperanto. Y la idea surge del hecho de poder contar con un producto de fácil colocación en el mercado y que genere los suficientes ingresos para crear una estructura capaz de crecer y fortalecerse en el tiempo de manera que el producto final llegue a cubrir las necesidades desde pequeñas tiendas hasta grandes conglomerados comerciales y llegue convertirse en todo un sistema de gestión empresarial.

# Investigación

Antes de asumir de arrancar el proyecto en su parte de desarrollo es conveniente hacer una investigación para determinar: las características que busca el usuario, las propiedades que tienen los sistemas más usados y el precio que el potencial usuario del producto está dispuesto a pagar por el mismo. En esta parte se expone estas tres premisas fundamentales de este proyecto. La mayor parte de esta sección está realizada sobre búsqueda en la web, fundamentalmente en la página softwareadvaice.com, de donde se extrae tanto las características de diferentes productos sino también el fundamento conceptual que dará la base al producto.

Es de hacer notar que al momento de estas notas iniciales no se ha definido el marco tecnológico a utilizar para el desarrollo.

## Características de un sistema POS.

De lo indicado por Justin Guinn[[1]](#footnote-1) se indica que las características más importantes de un sistema para manejo de una tienda son:

* **POS (*Point of Sales*)**: Ayuda al encargado del mostrador de la tienda en la atención de los clientes y su proceso de pago. Asegura al cajero el cobro de la cantidad correcta, ajusta los inventarios e imprime las facturas.
* **Control de Inventarios:** Automatiza la gestión de los niveles de inventario. Deduce el inventario cuando las ventas son completadas e indica alertas cuando los niveles de inventario alcanzan unos niveles de umbral predefinidos. Además provee de reportes sobre las tendencias de los movimientos de inventario.
* **Contabilidad de la tienda:** Centraliza los datos contables y suministra detalles precisos sobre el comportamiento del negocio. Incorpora cifras del inventario, así como otros gastos, tales como la nómina y el alquiler.
* **CRM (*Customer Relationship Managment*):** Almacena información de los clientes y su historial de compras. Posibilita a los administradores del negocio hacer un seguimiento de información de contacto, fechas claves tales como cumpleaños y aniversarios y los artículos preferidos para comercializar a los clientes.

A partir de estos componentes, se procede a definir en detalle cada uno de ellos, para así determinar su funcionalidad y su futura implementación.

POS: Es el sitio y momento donde se concreta una transacción de compra en una tienda. Es en este punto donde el cliente realiza el pago de las mercancías o servicios recibidos, además donde el comerciante prepara la factura al cliente y lo provee con las opciones de pago. Es por ello que el componente POS del sistema debe ser capaz de manejar todas las transacciones con el cliente, en cuanto a registro del cliente, recepción de los productos, facturación de los mismos, posibilidad de devoluciones, emisión de consulta de precios, recibir el pago según los diferentes medios que disponga y emitir la factura.

Control de Inventarios: En el nivel más básico, este componente ayudará al usuario a mantener con precisión los niveles del stock de productos. El objetivo principal de esta parte de la aplicación es advertir al usuario sobre los pedidos a realizar antes de que se agote la existencia. El tener niveles óptimos de inventario ayuda a capitalizar las oportunidades de ventas. Las características principales de este componente incluyen la clasificación de los elementos por departamento y/o tipo, establecer los umbrales mínimos y máximos y la generación de alertas cuando el stock esté por alcanzar tales niveles, funcionalidad de seguimiento por talla, color, estilo, entre otras características y la integración con otros módulos para la generación automática de, por ejemplo, órdenes de compra.

Contabilidad: Este módulo del sistema será el encargado de hacer el seguimiento de las cuentas por cobrar y por pagar, el valor actual del inventario y la depreciación del inventario. Su objetivo principal es centralizar todos los datos relacionados con la contabilidad y suministrar a los propietarios con cifras exactas, de manera que estos puedan tomar decisiones sobre el manejo de su negocio.

CRM (Administración de Relaciones con los Clientes): Los comerciantes más experimentados saben que es más rentable vender a clientes existentes que el mercadeo para hacer nuevos potenciales clientes. De allí la frase “el cliente es el rey”, es quién mantiene el negocio. Este componente va a ser el encargado de llevar lo concerniente a información de contacto, compras realizadas, artículos de interés en el futuro, fechas claves como cumpleaños o aniversarios y más. Con esta información en mano los usuarios del sistema cuentan con una herramienta que les ayudará a mantener y hacer crecer su activo más valioso, lo cual es su base de clientes.

### ¿Qué desean los clientes?

El mismo sitio softwareadvice.com, cuenta con una sección de evaluación de los productos, y allí los usuarios del mismo colocan sus opiniones. Las mismas están clasificadas en:

* Facilidad de uso
* Funcionalidad
* Calidad del producto
* Soporte al cliente

En cuanto a facilidad de uso, se pueden observar como intuitivo, bien explicado, cantidad de detalles, facilidad de aprendizaje por los cajeros, niveles de personalización, auto explicativo. Estas son premisas que hay que tomar en cuenta al momento del diseño.

El mercado de retail es muy extenso y tener un producto que satisfaga las necesidades de todos es bastante difícil de lograr. Pensando en eso, es que el sistema debe tener la capacidad de adaptarse a muchas circunstancias, ¿cómo se va a lograr eso?, es muy temprano para saberlo, sin embargo creemos que si se puede hacer.

El producto debe ser de una muy alta confiabilidad, y en lo posible, contar con un sistema de ayuda que le permita al usuario resolver la mayor cantidad de problemas por si mismo.

El soporte al cliente es de gran importancia, se debe contar con una plataforma que permita brindar la ayuda requerida por los clientes en el momento que la necesiten de forma rápida y efectiva.

La mayoría de los productos revisados cuentan con páginas web con una explicación detallada de sus productos, demos, verticales que manejan, precios, posibilidades de servicios web o remotos, soporte técnico y de ventas, integración con hardware, entre otros.

Pero definitivamente, todos cuentan con los módulos mencionados al comienzo. Los productos revisados hasta el momento son: Visual Retail Plus, Lightspeed Pos, TouchBistro, AIMsi y SimpleConsign.

Afortunadamente cada uno de ellos detallan muy bien sus características, razón por lo cual permite tomar ideas de uno y otro y así ofrecer una solución bastante amplia.

### Breve Análisis de Algunos Productos

Sobre la base de esta investigación vamos a tratar de ver que tiene cada uno de estos productos, para así tomar ideas que puedan ser provechosas para el proyecto.

#### Visual Retail Plus:

Cuenta con una forma centralizada de ofrecer POS y Manejo de Inventarios. Puede ser configurada para varios segmentos: tiendas de ropa, de regalos, ferreterías, de artículos electrónicos, artículos de oficina, entre otros.

El control central permite activar o desactivar varias características del sistema a requerimiento del usuario, tales cambios ocurren de manera instantánea, sin necesidad de re arrancar el sistema.

El Inventario es controlado de forma estricta por una “Matriz de inventario”, lo cual permite al usuario hacer ajustes sin necesidad de acceder otros módulos, permitiendo la creación de órdenes de compra o manifiestos de entrega, cambiar listas de precio, imprimir códigos de barra y etiquetas, categorizar inventarios, y más. Posee una característica llamada “Close to Real Time” que asegura que los datos de ventas e inventario estén actualizadas de forma precisa, aún sin internet.

Incluye un módulo de reportes que suministra una variedad de vistas y análisis, con recomendaciones para márgenes, transferencias y órdenes de compra asegurando una operación transparente.

Su interfaz visual, está algo recargada por lo que podría confundir en un inicio al usuario.

Posee la siguiente funcionalidad:

* Inventarios
* Punto de Ventas
* Órdenes de Compra
* Reportes
* Seguridad
* Control de Asistencia
* CRM

Cada uno de estos módulos está bien detallado en su página web

#### Lightspeed:

Su oferta está dividida en tres tipo de segmentos: retail, restaurantes y e-commerce. Su producto estrella se llama Lightspeed OnSite, el cual es exclusivo para Mac y IOS. Ofrece un sistema de inventario unificado, manejo de clientes y profundo sistema de análisis.

Su fuerte es el acceso desde donde sea, cuando lo requiera usando sus diversas interfaces para dispositivos Mac, IPad, y/o Iphone.

Posee capacidad de manejar multitiendas, con sincronización de los diversos inventarios.

La solución retail, ofrece:

* Manejo de inventarios
* Un amplio módulo de reportes
* Módulo de presupuestos, manejo de órdenes y facturación
* Control de asistencia

Cada una de estas funcionalidades no está muy detallada, pero da una idea de lo que ofrecen.

La solución para restaurantes es tal vez más desarrollada que el retail, al menos posee más detalle de funcionalidad. En su página web se observa las diferentes soluciones que ofrece para cada tipo de restaurant.

#### TouchBistro:

Es otra solución basada en el IPad, posee dos versiones: Full Service, para restaurantes, cervecerías y night clubs y Quick Service, para panaderías, cafes, camiones de comida y los llamados “fast casual”.

Sus características las tiene agrupadas de la siguiente manera:

* Personal y Gerencia
* Administración, personalización y Seguridad
* Reportes y Administración remota

Cada una de ellas están explicadas desde el punto de vista de lo que le muestran al usuario, con un poco de explicación, pero son interesantes de tomar en cuenta a la hora de ver sus propiedades.

#### SimpleConsign:

Solución basada en web, contiene módulos de POS, Manejo de Inventarios, CRM, pagos por partes, entre otros. Este POS está dirigido a pequeños negocios, tiendas de consignación, galerías de arte y tiendas de antigüedades. Su fuerte es la interfaz gráfica, que lo hace bastante intuitivo.

AIMsi (TRI-Tech):

Esta solución ofrece como oferta básica los módulos de POS, administración de inventario, CRM y contabilidad básica. Además ofrece los siguientes módulos Seguimiento de Reparaciones y servicio, Arrendamiento de corto plazo, Contratos de Compra y de Arrendamiento,

# Funcionalidad

De la investigación realizada se desprende que la funcionalidad de los sistemas POS va a depender de la vertical, aunque se puede partir de un genérico y desde allí, generar las diferentes verticales a atacar como producto. Básicamente se debe contener, como es referido al principio por cuatro módulos principales:

* POS
* Control de Inventarios
* CRM
* Contabilidad básica

Para cada uno de estas funcionalidades básicas, simplemente nos basaremos en la investigación previamente realizada.

## POS:

El POS, cubre una serie de funcionalidades que agrupadas constituyen el módulo POS del sistema, a continuación las enumeramos:

* Ventas
* Devoluciones
* Registro y Administración de Tarjetas de Regalo
* Descuentos
* Búsqueda de precios por SKU (número de referencia), grupo de proveedores, productos, estilos, talla, colores, etc.
* Manejo de punto y descuentos para compradores frecuentes
* Tarjetas de promoción
* Descuentos por cantidad o porcentaje
* Mezclas y coincidencias (productos alternos)
* Múltiples niveles de precios por Cliente, Cantidad y atributos de los artículos
* Métodos de entrega y despacho
* Pagos y reembolsos
* Reconciliación de la caja
* Suspensión (colocar en espera) y reanudar un número ilimitado de recibos
* Transferencias entre tiendas
* Final del día X y Z
* Órdenes especiales
* Comisiones de los representantes de ventas por cajero, grupo, vendedor, etc
* Integración PCI con tarjetas de débito y crédito
* Reparto de comisiones de los representantes
* Manejo de impuestos y exentos
* Diseño del recibo incluyendo logo y publicidad
* Seguimiento serializado del inventario y las transferencias
* Jornal electrónico
* Impresión del endoso de cheques
* Operaciones de emergencia fuera de línea
* Impresión de presupuestos
* Manejo de IVA y otros impuestos
* Muestra gráfica de los artículos de ventas
* Operaciones de Touch-screen
* Muestra de los Costos de Ventas Asociados a los Bienes
* Invalidación de pre y post ventas
* Manejo de cupones de descuento por cantidad, porcentaje, soporte para una variedad de reglas de cupones. Cupones del fabricante y de la propia tienda
* Notas de recibo pueden ser agregadas durante o después de finalizada la impresión del recibo de Regalo
* Múltiple selección de dispositivos de impresión, seleccionables al momento
* Impresión del Precio Sugerido por el Fabricante (MSRP) (opcional)
* Diferentes políticas de impresión dependiendo del tipo de transacción: venta, abono, devolución, etc.
* Permiso para vender artículos inválidos (fuera de stock, sin existencia, etc)
* Requerimiento de información del cliente basado en la oferta o tipo de transacción
* Consolidación de artículos
* Ventas por departamento
* Administración del número de recibo
* Almacenaje de créditos
* Diversos código de productos universales (UPC)
* Comenzar una transacción en una caja y terminar en otra
* Monitor de referencia
* Diseño de certificado de regalo
* Mantenimiento / cambio de los abonos existentes
* Pagos parciales para órdenes especiales, anulación o conversión de créditos de la tienda
* Planificación de descuentos por fecha, por porcentaje, cantidad de dinero o precio fijo.
* Cambio de precios permanentes por fecha
* Cuentas por cobrar
* Soporte para balanzas
* Envío de recibos por e-mail a los clientes

Esta es la funcionalidad soportada por el módulo de POS del Visual Retail Plus, alguna de ellas, tal vez, no apliquen al mercado nacional, sin embargo hay que chequearlas bien y de manera de desarrollar todas las que apliquen y hacerlo de tal modo que sean configurables por un instalador o el propio usuario.

## Control de Inventarios

Al igual que el POS la funcionalidad básica la vamos a tomar del módulo de inventarios del Visual Retail Plus. Allí aparecen las siguientes funcionalidades:

* Administración de precios
* Administración de Almacenes, matriz de color, talla, ancho, largo, medidas
* Diseñador de etiquetas de códigos de barra
* Campos definidos por el usuario
* Definición de artículos como inventarios o servicios
* Muestra de imágenes de artículos, imágenes por artículos o color
* Diversos código de productos universales
* Niveles mínimos y máximos (por temporada)
* Soporte para dispositivos de toma de inventarios físicos, cambio de precios, recepciones, transferencias, retornos al fabricante y registro de regalos
* Impresión de códigos de barra con múltiples diseños
* Impresión de etiquetas de estante
* Kits de inventario (kit de ventas y kits del fabricante)
* Copia de artículos (como plantilla) para creación de artículos nuevos
* Inventario físico
* Respaldo y recuperación
* Combinación de artículos y sus historia
* Manejo de inventarios por grupos (Departamentos), sub grupos (clase) y sub sub grupo (sub clase)
* Grupos combinados
* Compre uno y obtenga uno gratis
* Ventas de artículos enlazados
* Seguimiento del costo promedio
* Transferencias entre almacenes usando un manifiesto
* Búsqueda de almacenamientos remotos
* Conversión de ventas inválidas en válidas (después que el artículo ha sido definido)
* Duplicar un inventario para un nuevo almacén

## CRM

El módulo de CRM (Customer Relationship Management) de la aplicación Visual Retail Plus, está compuesto de las siguientes funcionalidades:

* Administración de contactos
* Hábitos de compra
* Pérdida de clientes
* Administración de e-mail masivos
* Seguimientos de cumpleaños por email, o impresión de etiquetas
* Combinación de clientes similares (por coincidencia de direcciones o nombres)
* Almacenamiento de exoneración de impuestos
* Históricos de compras de clientes y cuentas por cobrar
* Asignación de líneas de crédito a los clientes

Contabilidad Básica:

La contabilidad no es el fuerte de este sistema, por lo que hay que buscar información en otras fuentes. Así la tomamos de AIMsi y de Peachtree. La funcionabilidad típica que debe tener todo aplicación contable es:

* Cuentas por pagar
* Cuentas por cobrar
* Reconciliación bancaria
* Catálogo de cuentas
* Valoración del inventario
* Depreciación de los activos
* Reportes de toda la actividad contable

La tarea principal será la de compilar esta información para generar la funcionalidad básica que tendrá el proyecto Ekflugo, considerando como punto de partida la funcionalidad básica encontrada en la investigación realizada.

Está funcionalidad debe ser convertida en Casos de Uso y generación de Entidades, para ser el punto de partida de la realización del sistema.

## Módulos de Soporte

El sistema a diseñar no contará exclusivamente con los módulos cuya funcionalidad se ha descrito anteriormente, también debe contar con otros módulos que permitan brindar al usuario las características que espera de seguridad y robustez.

ES por ello que el sistema debe contar con los siguientes módulos:

### Seguridad:

Este módulo deberá administrar todo lo relacionado con el acceso al sistema: definición de perfiles de usuarios, grupos de usuarios, usuarios y permisos de acceso a las diferentes funcionalidades.

### Respaldo y Recuperación:

El mecanismo por el cual el usuario programará los respaldos periódicos de sistema y en caso tal de alguna eventualidad la recuperación desde un respaldo reciente.

### Configuración y Personalización:

Es muy difícil complacer a todos los usuarios de un sistema en cuanto a la apariencia del mismo, es por ello que este módulo debe poder adaptar la apariencia del mismo (GUI) al gusto del usuario, en cuanto a selección de colores, uso de logo y publicidad.

Además debe poder contar con la facilidad que el usuario con perfil administrativo pueda agregar usuarios, su perfil y permisos de acceso; configurar nuevas estaciones de trabajo, impresoras, balanzas y otros dispositivos de modo que tal administrador dependa lo menos posible del soporte.

## Requerimientos generales del Sistema:

El sistema a diseñar y desarrollar debe partir de las siguientes condiciones:

* De fácil uso por los usuarios: cajeros, vendedores, cobradores, almacenistas y administradores
* De agradable aspecto visual
* Con seguridad de acceso
* Preferiblemente multiplataforma (Windows, Linux, Mac)
* Uso de base de datos relacional
* De fácil instalación
* De fácil mantenimiento
* De fácil actualización
* Procesos de respaldo y recuperación
* Posibilidad de múltiples accesos por el usuario final (local y/o web)
* Diversos dispositivos de visualización: local, tablet o Smartphone
* Multiusuario
* Multimoneda
* MultiEmpresa
* Multilocalidad
* Posibilidad de adaptar varios idiomas o expresiones locales
* Integración con e-commerce

Al pensar en todas estas condiciones, surgen varias interrogantes para el diseño de la solución:

* ¿Será un producto único, con opciones de configuración?
* ¿Será un producto con varios niveles de oferta?, ¿básico, mediano y empresarial?
* ¿Qué nivel de precios lo hará competitivo en el mercado actual?
* ¿Será posible su exportación?

Por supuesto desde el punto de vista de la programación del sistema este se basará en Java, por ser el lenguaje que el autor domina, sin embargo y dadas las tendencias actuales de los sistemas hay que incorporar una serie de tecnologías de manera de poder hacer un producto robusto y de fácil actualización, para evitar su futura descontinuación.

En lo posible se usarán Patrones de Diseño estándares para facilitar su mantenimiento, así como herramientas de fácil obtención en la web. Se investigará la forma de agregar módulos de forma transparente al usuario, que permita la adaptación del sistema a los diversos requerimientos. Es posible que exista la necesidad de realizar diversas versiones dependiendo de la vertical a la que estaría dirigido el producto: restaurantes, tiendas de ropa, artículos deportivos, librerías, pequeños comerciantes, panaderías, cafés, abastos, supermercados, ferreterías, ventas de repuestos diversos, negocios de servicios y reparación, zapaterías, tiendas por departamentos, etc. Tal vez lo conveniente es identificar por el tipo de negocio y crear un perfil empresarial de modo de contar con los aspectos básicos y así sólo crear una base común y hacer las adaptaciones a nivel del aspecto visual.

Por ser un sistema pensado inicialmente al área de retail, no se ha incluido en el presente documento el área de manufactura ya que tiene unas implicaciones diferentes al de retail. En un futuro es posible que el producto pueda crecer hasta convertirse en un ERP integral, donde el módulo e POS sea uno de los componentes del sistema.

# Diseño

La etapa de diseño se compone de varias partes:

* Diseño Visual
* Diseño del catálogo de datos
* Diseño funcional

Dada la experiencia en anteriores y recientes proyectos, y de acuerdo con la metodología CD (ver siguiente sección), lo mejor es ir hacer un desarrollo por ciclos completos e ir despachando entregables funcionales del proyecto de tal manera que el proyecto vaya creciendo a medida que se vaya entregando y a la vez cada entrega represente una parte completamente funcional del sistema, lista para su uso.

La primera etapa del proyecto estará dirigida a una estación de un único usuario que concentre toda la funcionalidad del sistema, en una escala tal que permita su actualización a mayor cantidad de dispositivos/usuarios/localidades. Aun siendo única va a contar con un Front-End, es decir la parte operativa diaria de la tienda y el Back-End, desde donde se configurará la parte administrativa.

El Front-End corresponderá al trabajo diario del POS: apertura de la caja, registro de clientes, facturar, recibir pagos, emitir recibos/facturas, cierre de caja y tal vez emitir algún reporte que ese nivel amerite. El Back-End estará constituido por la parte administrativa del sistema, donde un usuario previamente identificado y autenticado, pueda crear usuarios, ingresar mercancía al inventario, crear/modificar listas de precios y obtener reportes diversos, entre otras actividades.

El patrón de diseño MVP (Model-View-Presenter) es la selección adecuada para este proyecto al permitir separar funcionalmente cada componente del sistema. Este patrón de diseño se deriva del conocido MVC. El objetivo principal del MVP y sus variantes es la separación de intereses entre la interfaz de usuario (UI), el modelo (datos de la aplicación), y el presentador (el controlador que maneja la lógica de presentación y las reglas del negocio) y, aunque existen varias implementaciones, se puede describir de la manera siguiente:

El **modelo** es una interface que define los datos a ser mostrados o actuar sobre la interfaz del usuario. El modelo desconoce por completo tanto a la vista como al presentador, su única función es almacenar y proporcionar datos.

La **vista** es una interface que muestra los datos (el modelo) y que enruta los comandos del usuario (eventos) al presentador para actuar sobre esos datos. Usualmente la vista tiene una referencia a su presentador. En la vista NO DEBE haber código de lógica de la aplicación. La vista desconoce por completa la existencia del modelo.

El **presentador** actúa sobre el modelo y la vista, es como un puente. Obtiene datos de los repositorios (el modelo) y los formatea en la vista y viceversa. Se podría decir que el presentador es el “hombre del medio” (ejecutado por el Controlador en el MVC) y que tiene referencias tanto al modelo como a la vista. Normalmente el presentador observa el modelo para recibir actualizaciones de datos. Cuando una actualización ha ocurrido en el modelo, actualiza la vista invocando el correspondiente método en la vista.



**Figura 2**. Representación esquemática del patrón MVP.

Al separar la parte de vista del resto del sistema, así como el modelo, se podría entre otras cosas, incorporar futuras tecnologías de presentación de la aplicación y esto hacerlo de forma muy suave. El tener los datos en una capa separada, los independiza de la base de datos a usar, pudiendo inclusive usar varias, por ejemplo una en las unidades de cajero, para funcionamiento local y otra en el servidor para funcionamiento centralizado.

Además este patrón se apoya en otro patrón de diseño, el patrón Observador. En este caso existen dos entidades: El Observador y los Observables, en este patrón el Observador está pendiente de los cambios realizados en los Observables y tomar acciones al respecto. De aquí nuestro Presentador será el Observador, que estará pendiente de los Observables, los cuales serán la Vista y el Modelo.

### Diseño Visual

El diseño visual es crucial para el sistema, los clientes actuales se han acostumbrado a ver excelentes aplicaciones en diversos dispositivos desde el escritorio hasta sus teléfonos inteligentes. La primera etapa del proyecto será desarrollada para un ambiente de escritorio, esto debido a la incorporación de dispositivos externos que deben ser manejados por un cajero, tales como impresoras fiscales, pantallas auxiliares y/o balanzas.

Aunque el framework SWING es muy usado para aplicaciones de escritorio, se estima que el nuevo **JavaFX2** ha alcanzado un buen grado de madurez como afrontar este proyecto: pensado desde el principio para hacer uso del patrón MVP; sus posibilidades de formateo gráfico usando hojas de estilo (CSS); su independencia del diseño de la funcionalidad, lo hacen ideal para tener una aplicación muy atractiva desde el punto de vista visual.

El diseño visual estará enmarcado dentro de la parte View del patrón MVP.

### Diseño del Catálogo de Datos

Los datos serán modelados en la base de datos usando la herramienta **MySQLWorkbench**, ya que de forma gráfica podemos observar cada tabla de datos, y a partir del modelo Entidad– Relación (E-R) resultante generar la base de datos. Por otro lado, usando el framework **JPA** (Java Persistence API) podemos aprovechar el mismo diseño para hacer que nos sirva para otras bases de datos relacionales tales como, por ejemplo **Derby**, como base de datos local del POS. El poder hacer que el sistema pueda tener acceso a varias bases de datos hace que el sistema pueda crecer desde un nivel mono-usuario hasta un sistema multi-usuario, multi-localidad, en otras palabras nivel empresarial.

El diseño del catálogo de datos estará enmarcado dentro de la parte Model del patrón MVP.

### Diseño de Funcional

Aunque se mencionó en la parte del modelo, el JPA sería más adecuadamente posicionado

## Metodología

Además se deben seleccionar la metodología y herramientas a trabajar, de manera que el sistema vaya generando entregables de forma continua y no esperar a que esté completamente listo para comenzar a realizar pruebas.

La metodología Scrum está orientado a ese fin, sin embargo está pensado para equipos de desarrollo. La metodología Kanban, parece más adaptada a las condiciones actuales del proyecto, donde se arrancará con un solo desarrollador. Además existe el concepto de Continuos Delivery, con el que se generan versiones entregables con cierta frecuencia, permitiendo contar con la herramienta en un ciclo de desarrollo tal vez más corto.

Con esto en mente existen herramientas que se pueden usar las cuales facilitan la creación y seguimiento de tareas en las diferentes etapas del proyecto, lo cual permite llevar un control bastante preciso del sistema a desarrollar.

En el siguiente diagrama se muestra la filosofía de Entrega Continua (CD), en la que basaremos el proyecto:



**Figura 1.** Diagrama de Secuencia de la Metodología Continuos Delivery

En los momentos iniciales del proyecto, esta metodología se realizará de forma manual, una vez que el proyecto avance y en la medida que vaya creciendo se utilizará una herramienta destinada a la automatización de este proyecto. Dadas las características encontradas en el proyecto **Jenkins**, esta será la herramienta a utilizar.

Una de las exigencias de la metodología CD es contar con un sistema de control de versiones. El sistema de control de versiones permitirá contar con las diferentes versiones en sus diferentes etapas: desarrollo, prueba y reléase. Existen varios sistemas que permiten llevar a cabo el control de versiones, para este proyecto se ha seleccionado **Github**, por conocer de antemano su funcionamiento. Este sistema se puede instalar localmente pero cuenta con un sitio en la web, gratuito, que nos permite que sea de fácil acceso desde cualquier sitio. La contraparte de su gratuidad es que de acceso público y cualquiera que conozca su ubicación en la red lo podría descargar. Para ello al sistema Ekflugo, se le dará un nombre clave de manera de hacerlo invisible a los que deseen buscar información en Github.

Para mantener el control de los equipos de desarrollo, prueba y liberación se ha determinado la necesidad de un sistema de control de proyecto. Tal como lo mencionamos anteriormente este proyecto será regido por la metodología Kanban. La investigación correspondiente ha dado como fruto un varias herramientas basadas en Kanban: **Trello**, **Kanbanchi** y **Kanbanflow**. Actualmente de las tres se ha usado la primera, las otras dos no se han evaluado. Las tres herramientas son de uso gratuito, teniendo la posibilidad de agregar diversas funcionalidades las cuales algunas son gratis y otras pagas. Trello tiene la particularidad que está pensada como una herramienta de uso general. Las dos siguientes deben ser evaluadas para escoger la que mejor se adapte a los requerimientos del sistema.

A continuación se hará una evaluación de las tres herramientas para seleccionar la más adecuada.

1. <http://www.softwareadvice.com/retail/?deployment_id=&market_products_sort_order=&market_products_sortby=nb_reviews&more=true&price_ranges=&stars=4&segment_id=&platforms=&int_site_code=&layout=var_c0&size_id=> [↑](#footnote-ref-1)