



**Formato de planos**

**FP-02**



<b>Nombre del caso:</b>	<b>Necesidad</b>	<b>ID:</b>	<b>01</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>

Amazonic Guatemala es una empresa dedicada a la venta de productos, y por esto, necesita un programa intuitivo que le ayude a facilitar la venta de productos en varias tiendas, generando reportes de ventas, gastos entre otros.





<b>Nombre del caso:</b>	<b>Idea</b>	<b>ID:</b>	<b>02</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>

Por la necesidad de no ser tan tedioso el saber que se tiene en stock cuando un cliente desea hacer una orden, y hacer que una persona este revisado, además de ser muy molesto para cliente, es demasiado tardado y poco eficiente.

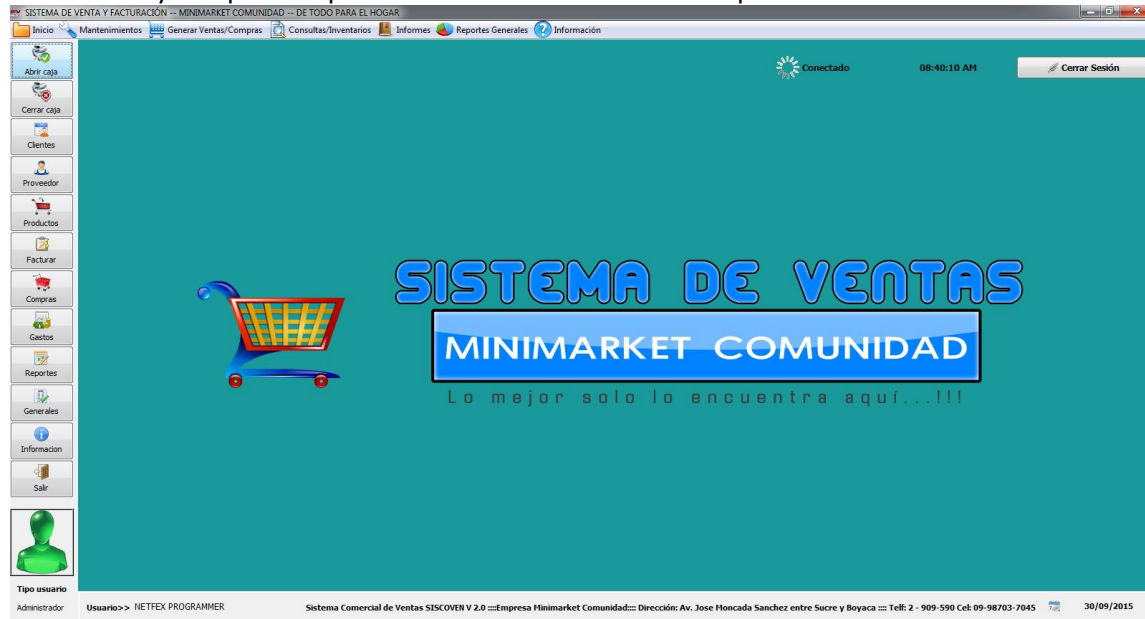
Con la experiencia de los trabadores, es recomendable tener una aplicación que les permita hacer una orden y que automáticamente busque en una base de datos si se posee el producto.





<b>Nombre del caso:</b>	<b>Desarrollo de la Idea</b>	<b>ID:</b>	<b>03</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>

La aplicación MiniMarket permitirá venta de productos, todo tipo de reportes con sus respectivas estadísticas y un apartado para extender la factura de la compra.





<b>Nombre del caso:</b>	<b>Métricas(Core de la Aplicación)</b>	<b>ID:</b>	<b>04</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>

La interfaz siguiente, es del apartado de ventas, donde el trabajador podrá ir añadiendo productos a vender y cuando este quiera cancelar se creará una factura automáticamente.

Ventas

**FORMULARIO DE VENTAS**

Número

7

Fecha

27/05/2013

Hora

13:58:59

**Datos de Funcionarios**

ID	Nombres	Apellidos
1	Carlos Ariel	Chávez Madiel

**Datos de Cliente**

ID	Descripción	RUC
1	Casual	-

**Datos de Artículo**

ID	Descripción	Precio de Venta	Stock	Cantidad	Total
4	Tedado Mtek español	50000	1		

Agregar
 Eliminar

ID	Artículo	Precio Venta	Cantidad	Total
1	Monitor LCD LG 20"	549000	1	549000
2	Impresora HP 4500 Color	448000	1	448000
4	Teclado Mtek español	50000	1	50000
5	Webcam Genius DC1	49000	1	49000
6	Hoja Oficio Pack	26600	2	53200

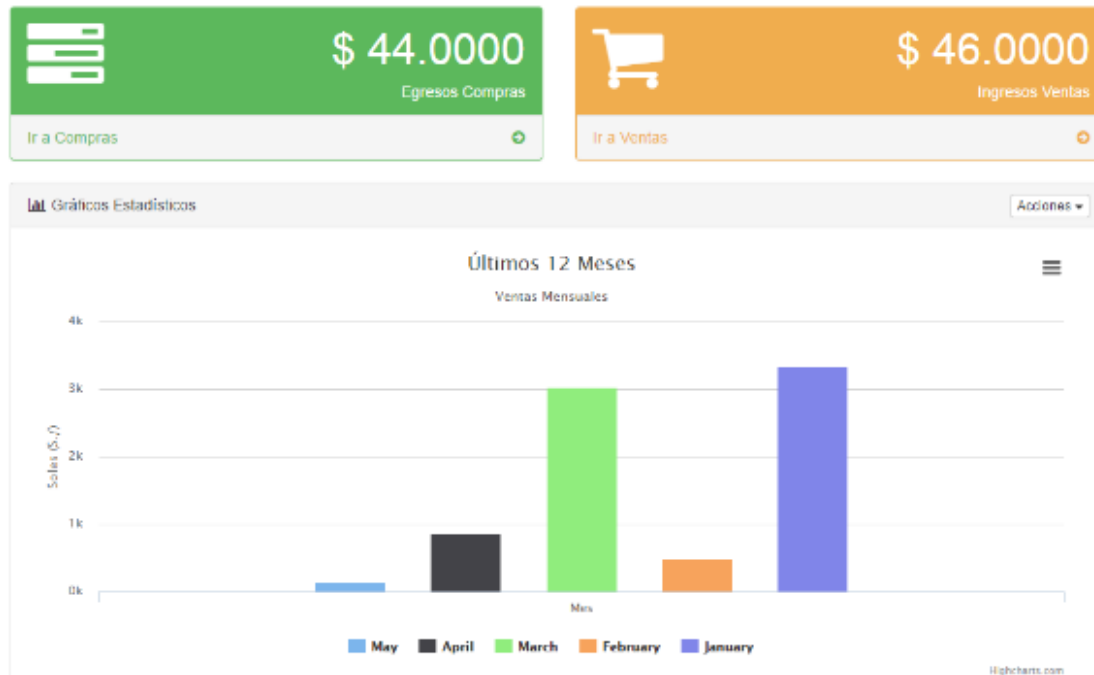
Nuevo
 Guardar
 Cancelar

1149200



<b>Nombre del caso:</b>	<b>Métricas(Core de la Aplicación)</b>	<b>ID:</b>	<b>05</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>

La interfaz siguiente, es del apartado de reportes y estadísticas, donde el trabajador podrá acceder a un registro de eventos de la venta de cada día, así mismo como una estadística de los productos mas vendidos a la semana.





<b>Nombre del caso:</b>	<b>Métricas(Core de la Aplicación)</b>	<b>ID:</b>	<b>06</b>
<b>Autor</b>	<b>Carlos Hernandez</b>	<b>Fecha Creación:</b>	<b>22/02/17</b>
<ul style="list-style-type: none"> <li>Las métricas a utilizar para obtener los productos más vendidos son las siguientes: <ul style="list-style-type: none"> <li><b>Característica 1:</b> Cantidad</li> <li><b>Característica 2:</b> Precio</li> <li><b>Característica 3:</b> Fecha de caducidad</li> </ul> </li> <li>Para poder determinar los productos más vendidos, se procederá a tomar la cantidad de productos que se vendieron en un día, el corte se hará a una hora específica para evitar la venta de productos y que estos no sean tomados en cuenta.</li> <li>El precio hará un juego importante a la hora de realizar el algoritmo, ya que lo que le interesa a la empresa son las ganancias, es decir, que nos interesa que productos se han vendido más, pero cuales a su vez han generado más ganancias</li> <li>La fecha de caducidad es importante, obtendremos una estadística aproximada de que productos se han vendido con más frecuencia, si están o no cerca de su fecha de vencimiento.</li> <li><b>Razón de existir:</b> <ul style="list-style-type: none"> <li>Esta aplicación aumentara exponencialmente el crecimiento de la empresa no solo haciendo sus procesos muchos más eficientes, si no también que los clientes se darán cuenta la clase de atención que pueden recibir en estos locales, siendo rápidos, concisos y con una excelente calidad de servicios.</li> </ul> </li> </ul>			



Nombre del caso:	Listado de Funcionalidades	ID:	06
Autor	Carlos Hernandez	Fecha Creación:	22/02/17
<ul style="list-style-type: none"><li>• Módulos extras que permitirán una mejor experiencia con la aplicación:<ul style="list-style-type: none"><li>○ La aplicación se adapta a cualquier forma de pago del cliente.</li><li>○ Aplicación de descuentos definidos por la persona encargada.</li><li>○ Acceder al inventario, buscar, editar o eliminar productos.</li></ul></li></ul>			



## ANEXOS

### 1. DESCRIPCION DEL PROBLEMA

Amazonic Guatemala es una empresa dedicada a la venta de productos, y por esto, necesita un programa intuitivo que le ayude a facilitar la venta de productos en varias tiendas, generando reportes de ventas, gastos entre otros.

La empresa se ha visto afectada increíblemente por la decadencia de los clientes, el poco control que se puede tener sobre un stock, ya que el papel es demasiado sencillo alterar por una persona ajena al departamento encargado del control de productos, que los despachadores aplican mal un descuento provocándole pérdidas a la empresa, o que simplemente se les olvide cerrar una cuenta de algún cliente y este pueda extraer del local la mercancía sin haberla cancelado en caja.

### 2. DEFINICION DE LOS REQUIRIMIENTOS DEL SISTEMA

- El sistema deberá, a petición del operador, agregar productos para su venta
- El sistema generara facturas por cada venta, y guardar en un registro la venta del día.
- Generará estadísticas acerca de los productos
- El sistema ejecutara descuentos personalizados por el operador.
- El sistema será adaptable a cualquier producto y/o forma de pago.
- El sistema no tardar más de 5seg en ejecutar una consulta al stock, y devolver si existe o no el producto para su venta.
- El sistema se desarrollará en un entorno Java EE.
- El sistema deberá utilizar la autorización de la persona encargada para ejecutar operaciones como editar el stock.
- El sistema podrá almacenar gran cantidad de productos sin dificultad, a su vez compras seguidas sin ningún tipo de problema.
- El sistema automáticamente realizara cortes a una hora determinada por el usuario, posteriormente estos datos serán utilizados para hacer una estadística.





- El sistema tendrá la capacidad de poder eliminar o detener una venta y no alterar el stock por los inconvenientes.
- El sistema tendrá un límite de consultas por minuto, por motivos de seguridad.

### 3. PLANTEO INICIAL DE LA SOLUCION

Por la necesidad de no ser tan tedioso el saber que se tiene en stock cuando un cliente desea hacer una orden, y hacer que una persona este revisado, además de ser muy molesto para cliente, es demasiado tardado y poco eficiente.

Con la experiencia de los trabadores, es recomendable tener una aplicación que les permita hacer una orden y que automáticamente busque en una base de datos si se posee el producto.

El programa tendrá que tener un nivel de gran seguridad, por que tendrá el corazón de la empresa en sus bases de datos, se tendrá que restringir la manipulación del mismo por personas no calificadas, es cierto, el sistema que se quiere implementar en esta empresa, tendrá unos grandes beneficios, pero también grandes riesgos, es por esto que como desarrolladores del mismo es nuestro deber, el garantizar que el programa será funcional en su totalidad, pero también que poseerá una seguridad alta, difícil de penetrar, y que siempre tendrá un resguardo de información por si algo inoportuno llegara a pasar.

Las ventajas que poseerá este proyecto, es que con la tecnología que tenemos al alcance no podemos limitarnos a una sola computadora, el proyecto puede expandirse a cualquier dispositivo que la empresa proporcione para el control y manejo de toda clase de función que el programa les brinda, siendo un placer para el cliente que sea atendido como se merece y brindándole los mejores servicios posibles.

### 4. CICLO DE VIDA DEL SOFTWARE

#### ➤ Análisis:

Se necesitan cierta cantidad de datos para alimentar el sistema estos pueden ser 2.

- Datos de Entrada: venta y compra de productos, formas de pago, gastos e ingresos.
- Datos de Salida: reportes de ventas, estadísticas de los productos, facturas, total de ventas clasificadas por día y por mes, entre otros servicios.



➤ **Diseño:**

En esta fase se procede a diseñar algoritmos que satisfagan las necesidades ya descritas, cabe destacar que un algoritmo es un conjunto finito de pasos para resolver un problema, este puede ser escrito en pseudocódigo y para su mejor comprensión graficado en un diagrama de flujo.

El sistema será trabajado en una forma modular, es decir que cada proceso será individual, es decir que, si algún proceso por algún motivo no logra terminarse de ejecutar o no se ejecuta correctamente, las demás partes del sistema seguirán funcionando.

➤ **Implementación:**

La fase donde el sistema se empieza a testear con productos “falsos” para verificar su correcto funcionamiento, esto lo hará el tester del software y aun no se procede a instalarlo en la empresa.

➤ **Prueba e Integración:**

En esta fase el software está listo para ser puesto a prueba por la empresa, con datos reales, ejecutando las operaciones en tiempo real, posteriormente con un visto bueno, se procede a instalar el software y ponerlo en marcha.

➤ **Depuración:**

En esta fase suelen salir a luz los pequeños bugs que no fueron detectados en las fases anteriores, o los cambios ya sean estéticos o backend que desee la empresa, estos errores son corregidos antes de proceder a la completa implementación del software.

➤ **Mantenimiento:**

Esta fase es una recursiva ya que no se ejecutará solo una vez, es recomendado al menos cada 6 meses realizarle un mantenimiento o actualización al sistema, ya que puede suceder que algunos algoritmos estén obsoletos.



## DICCIONARIO DE CLASES

### Clase: Principal

Esta es la clase principal, encargada de todo el funcionamiento del sistema.

#### Atributos:

#### Metodos:

ValidarSesion	Devuelve un booleano si el inicio de sesión es correcto o no.
Menu	Posee el menú de acciones
CerrarSesion	Cierra la sesión del trabajador

### Clase: Auxiliar

Esta es la clase auxiliar, encargada de procesos para no cargar la clase principal

#### Atributos:

#### Metodos:

Vender	Devuelve un booleano si se hizo la venta correctamente, y ejecuta Facturas()
BuscarProducto	Devuelve un booleano si se posee en stock el producto escogido
Facturas	Esta es ejecutada luego de haber realizado la venta, y se encarga de guardar en los registros la venta y generar una factura
Comprar	Encargada de guardar en stock los nuevos productos
Inventario	Se encarga de editar el stock que se posee.
Estadísticas	Genera estadísticas de los productos vendidos con características escogidas por el usuario.
FormaDePago	Es la encargada de controlar que forma de pago , con que el cliente desea cancelar(chegue, tarjeta débito o crédito)
CancelarCompra	Se encarga de quitar la cantidad comprada del stock
VerFacturas	Muestra las facturas creadas anteriormente



## GLOSARIO DE TERMINOS

<b>Términos:</b>
<b>Algoritmo</b>
Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan y especifica el orden en que se ejecutan estas acciones. Los algoritmos ayudan al programador a planificar un programa antes de su escritura en un lenguaje de programación.
<b>Ámbito de clase:</b>
Las variables privadas definidas fuera de los métodos internos a la clase tienen ámbito de clase. Son accesibles desde todos los métodos del interior de la clase, con independencia del orden en que están definidas. Los métodos privados también tiene ámbito de clase.
<b>Análisis:</b>
Proceso de identificación, modelado y descripción de lo que hace un sistema y de cómo trabaja
<b>Aplicación:</b>
Programa autónomo Java tal como cualquier programa escrito utilizando un lenguaje de alto nivel. Las aplicaciones se pueden ejecutar desde cualquier computadora con un intérprete Java.
<b>Diccionario de Clase:</b>
Colección organizada de clases que proporciona un conjunto de componentes y abstracciones reutilizables
<b>Boolean:</b>
Tipos primitivos de datos en Java. El tipo boolean puede tomar sólo dos valores: true (verdadero) y false (falso).
<b>Clase:</b>
Colección encapsulada de datos y operaciones que actúan sobre los datos. El concepto de clase es fundamental en programación orientada a objetos. Una clase consta de métodos y datos. Los métodos de una clase definen el conjunto de operaciones permitidas sobre los datos de una clase (sus atributos). Una clase puede tener muchas instancias de la clase u objetos.
<b>Compilación:</b>
Proceso de traducción de un lenguaje de programación. Normalmente este proceso implica la traducción de un lenguaje de programación de alto nivel a lenguaje de programación de bajo nivel, o el formato binario de un conjunto de instrucciones específicas. La traducción se realiza con un programa denominado compilador. Un compilador java traduce los programas en bytecodes.
<b>Compilador:</b>
Programa de software que realiza un proceso de compilación (traducción del lenguaje fuente a lenguaje máquina) de un programa escrito en un lenguaje de programación de alto nivel. En el caso de Java, es un programa que traduce el código fuente Java en bytecode. El compilador de J2SDK se denomina javac.
<b>Constructor:</b>
Método especial utilizado para inicializar el estado de un nuevo objeto. El constructor permite crear objetos utilizando el operador new. El constructor tiene exactamente el mismo nombre que la clase que lo contiene. Los constructores se pueden sobrecargar con



el objetivo de facilitar la construcción de objetos con diferentes tipos de valores iniciales.
<b>Depuración:</b>
Proceso de encontrar, fijar y eliminar errores en un programa. Para estas tareas se suele utilizar una herramienta de programación conocida como depurador.
<b>Diseño:</b>
Actividad de definir como se debe estructurar e implementar un programa.
<b>Función:</b>
Construcción matemática a la que se pueden aplicar valores y que devuelve un resultado.
<b>Implementación:</b>
La actividad de escribir, compilar, probar y depurar el código de un programa.
<b>Interfaz:</b>
Una interfaz se trata como una clase especial de Java. Cada interface se compila en un archivo independiente de bytecode, tal como una clase ordinaria. No se puede crear una instancia de la interfaz. La estructura de una interfaz Java es similar al de una clase abstracta en la que se puede tener datos y métodos. Los datos, sin embargo, deben ser constantes y los métodos pueden tener sólo declaraciones sin implementación. En Java existe sólo herencia simple y una clase puede heredar de una superclase. Esta restricción se puede superar por el uso de una interfaz.
<b>Método:</b>
Una colección de sentencias que se agrupan juntos para ejecutar una operación.
<b>Objeto:</b>
Una instancia de una clase específica. En general, se puede construir cualquier número de objetos a partir de una clase.
<b>Programa:</b>
Un conjunto de instrucciones (o sentencias) que describen alguna aplicación o actividad ejecutada en una computadora.
<b>UML:</b>
Lenguaje unificado de modelado que proporciona notación estándar visual para documentar el análisis y diseño de sistemas orientados a objetos.