Instituto Tecnológico de San Juan del Río



PROYECTO CENCEL

CENCEL S.A. DE C.V.

**P RE S E N T A:**

**MAYRA IBARRA BADILLO 12590431**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

**JOEL IBARRA BADILLO 11590446**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**NATANAEL CRUZ BRAVO 14590196**

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

**LUIS FERNANDO NIEVES NIEVES 14590223**

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

PERIODO ENERO – JUNIO 2017

**ÍNDICE**

**[1.](#_Toc474414716)****[METODOLOGÍA](#_Toc474414716)** [3](#_Toc474414716)

**[2.](#_Toc474414717)****[FASES](#_Toc474414717)** [3](#_Toc474414717)

**[1ª Fase: Planificación del proyecto.](#_Toc474414718)** [3](#_Toc474414718)

**[2ª Fase: Diseño.](#_Toc474414719)** [4](#_Toc474414719)

**[3ª Fase: Codificación.](#_Toc474414720)** [4](#_Toc474414720)

**[4ª Fase: Pruebas.](#_Toc474414721)** [5](#_Toc474414721)

**[3.](#_Toc474414722)****[JUSTIFICACIÓN DE LA METODOLOGÍA](#_Toc474414722)** [6](#_Toc474414722)

**[4.](#_Toc474414723)****[ROLES XP](#_Toc474414723)** [7](#_Toc474414723)

**[5.](#_Toc474414724)****[CRONOGRAMA DE ACTIVIDADES](#_Toc474414724)** [8](#_Toc474414724)

**[6.](#_Toc474414725)****[ESQUEMA DE PANTALLAS](#_Toc474414725)** [9](#_Toc474414725)

**[7.](#_Toc474414726)****[SELECCIÓN DE COLORES](#_Toc474414726)** [17](#_Toc474414726)

# **METODOLOGÍA**

“XP: Extreme Programing”

Es Una metodología de desarrollo ligero (ágil) basada en una serie de valores y prácticas, que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

# **FASES**

**1ª Fase: Planificación del proyecto.**

**Historias de usuario:** Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

**Release planning:** Es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa.

**Iteraciones**: Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

**Velocidad del proyecto:** La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración.

**Programación en pareja:** La metodología X.P. aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.

**Reuniones diarias.** Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

**2ª Fase: Diseño.**

**Diseños simples:** Hacer el proyecto todo lo menos complicado posible para conseguir un diseño fácilmente entendible e impleméntale que a la larga costará menos tiempo y esfuerzo desarrollar.

**Glosarios de términos:** Usar glosarios de términos y un correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

**Riesgos:** Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.

**Funcionalidad extra:** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.

**Refactorizar.** Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es necesario refactorizar cuando se va a utilizar código ya creado.

**Tarjetas C.R.C.** El uso de las tarjetas C.R.C (Class, Responsabilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

**3ª Fase: Codificación.**

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

X.P afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se haga lo que se haga, aunque se añadan más desarrolladores y se incrementen los recursos. La solución que plantea X.P es realizar un nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

**4ª Fase: Pruebas.**

X.P hace uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

El uso de los test en X.P es el siguiente:

* Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
* Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.
* Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa.
* Los distintos test se deben subir al repositorio de código acompañados del código que verifican. Ningún código puede ser publicado en el repositorio sin que haya pasado su test de funcionamiento, de esta forma, aseguramos el uso colectivo del código (explicado en el apartado anterior).
* El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.
* Test de aceptación. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.
* Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos

# **JUSTIFICACIÓN DE LA METODOLOGÍA**

XP es una metodología que permite las mejoras continuas una tras otra y la planificación por etapas, a través del desarrollo iterativo e incremental, pruebas unitarias, la programación en parejas, simplicidad de código y la corrección de todos los errores antes de añadir nueva funcionalidad. Además de que permite el contacto frecuente con el cliente para su funcionalidad y desarrollo, al mismo tiempo que se formula un plan de entregas compuesto por iteraciones no mayores a tres semanas.

La metodología trabaja con historias de usuario, que describen la funcionalidad del sistema en partes, con las cuales se determinan los tiempos de desarrollo y prioridad de cada una, al mismo tiempo que ayudan a verificar que el sistema cumpla con lo requerido.

X.P además sugiere un modelo de trabajo usando repositorios de código dónde los programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones.

XP también cuenta con un modelo de desarrollo colectivo que permite que todos los programadores estén implicados en todas las tareas; es decir, que cualquiera de los integrantes pueda modificar alguna clase o método de otro programador, de esta manera todas las partes involucradas actúan y forman parte del proyecto, así mismo se asignan roles y tareas de programación a cada integrante.

# **ROLES XP**

**Programador**

Responsable sobre el código, responsable sobre el diseño (refactorización, simplicidad), responsable sobre la integridad del sistema (pruebas).

Joel Ibarra Badillo

Mayra Ibarra Badillo

Luis Fernando Nieves Nieves

Natanael Cruz Bravo

**Cliente**

Define especificaciones, influye sin controlar, define pruebas funcionales.

Raúl Raziel Landeros Pérez

**Encargado de Pruebas (Tester)**

Apoya al cliente en la preparación/realización de las pruebas funcionales, ejecuta las pruebas funcionales y publica los resultados.

Equipo programador => Pruebas Internas

Ing. Raúl Raziel Landeros Pérez => Pruebas externas

**Encargado de Seguimiento (Tracker)**

Recoge, analiza y pública información sobre la marcha del proyecto sin afectar demasiado el proceso, supervisa el cumplimiento de las estimaciones en cada iteración, informa sobre la marcha de la iteración en curso, controla la marcha de las pruebas funcionales, de los errores reportados, de las responsabilidades aceptadas y de las pruebas añadidas por los errores encontrados.

Ing. Raúl Raziel Landeros Pérez

**Entrenador (Coach)**

Responsable del proceso en su conjunto, identifica las desviaciones y reclama atención sobre las mismas, guía al grupo de forma indirecta (sin dañar su seguridad ni confianza), interviene directamente si es necesario, atajar rápidamente el problema.

Ing. Elsa Cuevas Carrillo

**Consultor**

Apoya al equipo XP en cuestiones puntuales.

Ing. Elsa Cuevas Carrillo

**Jefe del Proyecto**

Favorece la relación entre usuarios y desarrolladores, cubre las necesidades del equipo XP, asegura que alcanza sus objetivos.

Ing. Raúl Raziel Landeros Pérez

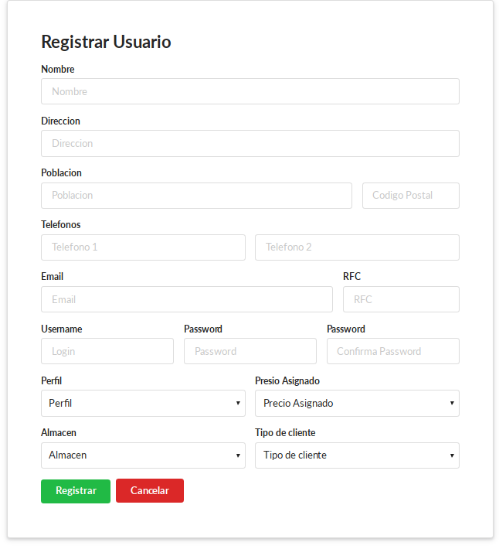
# **CRONOGRAMA DE ACTIVIDADES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actividades | | | | |
|  |  |  |  |  |
| **1** |  |  | Planificación del proyecto |  |
| **1.1** | |  | Historias de Usuario |  |
| **1.2** | |  | Plan de Publicaciones (cronograma de entregas) |  |
| **1.2.1** | | | Planificación de entregas (velocidad del proyecto |  |
| **1.3** | |  | Iteraciones |  |
| **1.4** | |  | Reuniones |  |
| **2** |  |  | Diseño |  |
| **2.1** | |  | Diseño Simple |  |
| **2.2** | |  | Glosario de términos |  |
| **2.3** | |  | Tarjetas CRC |  |
| **3** |  |  | Codificación |  |
| **4** |  |  | Pruebas |  |
| **4.1** | |  | Test de aceptación |  |

# **ESQUEMA DE PANTALLAS**

**Registrar Usuarios:**

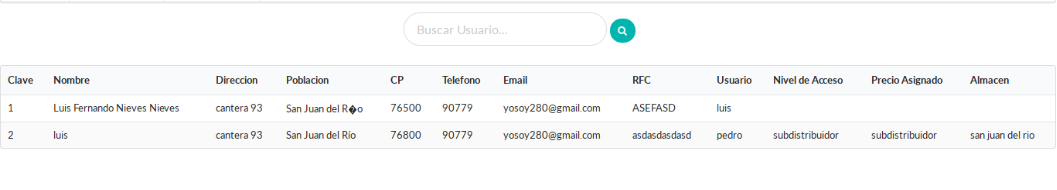
La pantalla de registrar usuario contiene los siguientes campos, los cuales deben ser llenados correctamente, la información debe ser validada para evitar redundancia o inconsistencia de los datos.



El botón Registrar sirve para guardar el registro en la base de datos, deben estar llenados todos los campos requeridos de lo contrario no podrá guardar el registro en la base de datos El botón Cancelar interrumpe la transacción y no se realiza la inserción en la base de datos y se redirige a la página principal.

**Buscar Usuario:**

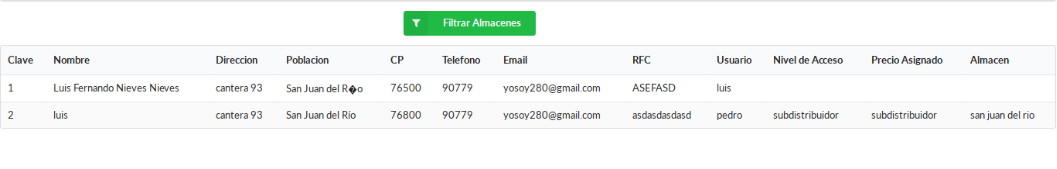
La pantalla de buscar usuario cuenta con una caja de texto, la cual requiere de un dato numérico, en este caso la clave del usuario.



Si el usuario existe mostrara la información del usuario, de lo contrario mostrara un mensaje notificando que el usuario no existe. La información que se muestra de este resultado puede ser utilizada únicamente para consulta.

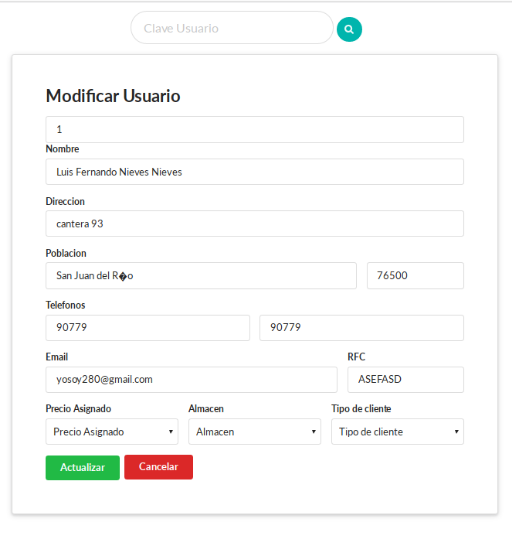
**Directorio:**

La pantalla directorio muestra todos los registros almacenados en la base de datos en base a los almacenes. Se puede aplicar un filtro mediante el almacén, sino se aplica ningún filtro, la consulta traerá todos los registros y solo podrán ser consultados.



**Modificar Usuario:**

La pantalla modificar usuario cuenta con un campo de texto, el cual requiere de una clave de usuario para poder realizar la consulta a la base de datos.



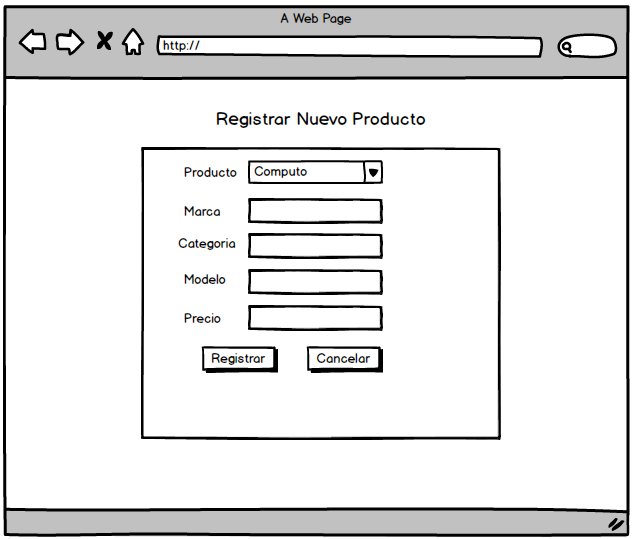
Si la clave del usuario no está registrada, se notificara, de lo contrario se llenara el formulario con la información correspondiente a la clave de usuario, la cual podrá ser modificada.

El botón actualizar finaliza la transacción de información, mientras que el botón cancelar interrumpe la acción y redirige a la página de inicio

**Registrar Producto:**

La pantalla registrar producto puede almacenar un solo tipo de producto, el cual tiene que ser seleccionado de la lista desplegable, una vez seleccionado se mostraran los campos correspondientes para su llenado.

Deben existir campos mínimos requeridos y validados, de lo contrario podría ocurrir algún error en la transacción.



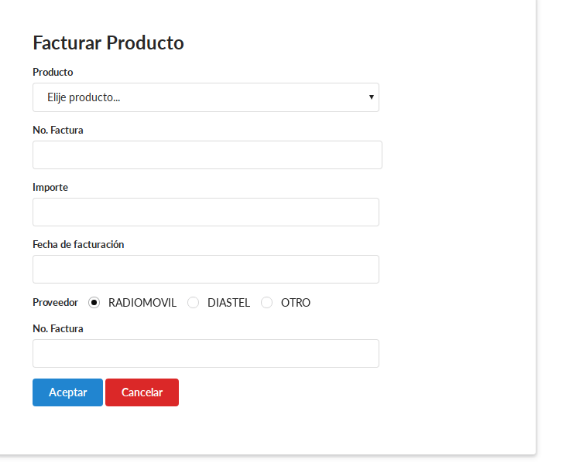
El botón Registrar completa la transacción e inserta el registro en la base de datos, mientras que el botón cancelar interrumpe la transacción y redirige a la pantalla principal

**Facturar Producto:**

La pantalla facturar producto requiere que se indique que tipo de producto se va a facturar, esto para mostrar los campos en base a las características de cada producto, así se podrá capturar la información correspondiente con cada producto.

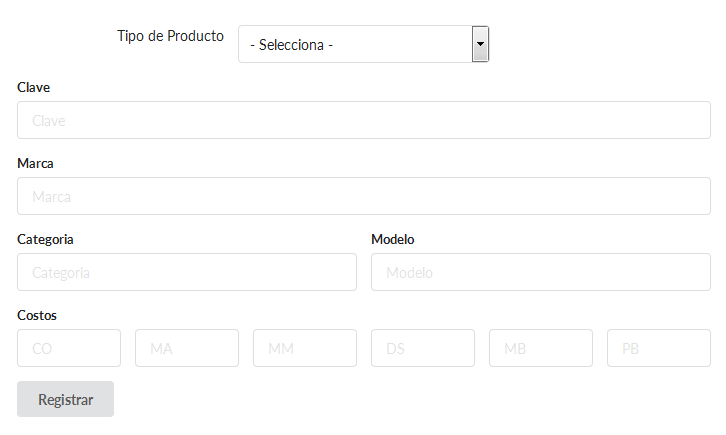
El proveedor debe ser seleccionado, si se elige otro, debe ser indicado en un campo de texto el cual aparecerá si se selecciona otro proveedor

El botón aceptar guarda el registro de la factura en la base de datos y el botón cancelar interrumpe la transacción.



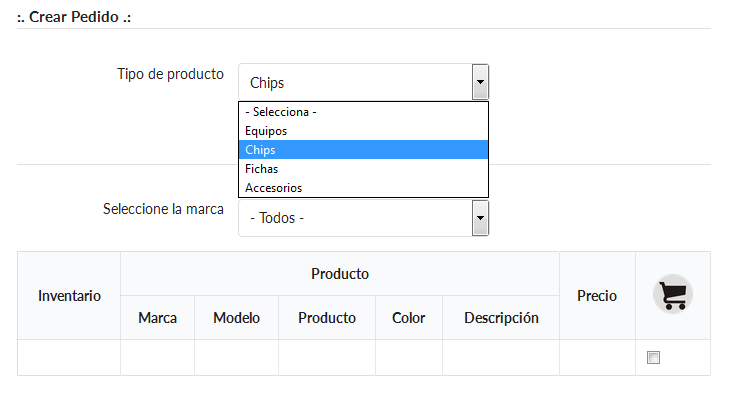
**Registrar Productos**

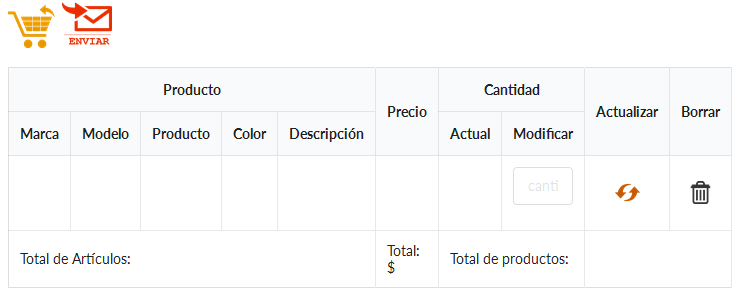
En esta pantalla se registran los nuevos productos, seleccionando desde el tipo.



**Crear Pedido:**

La pantalla de crear pedido permite seleccionar entre los distintos tipos de productos que manejan en Cencel, y hasta seleccionar el tipo de producto aparece el inventario con los productos existentes y una casilla para seleccionar los deseados, que se almacenan en un carrito de compras, donde se puede cambiar la cantidad de productos o eliminar y enviar el pedido.

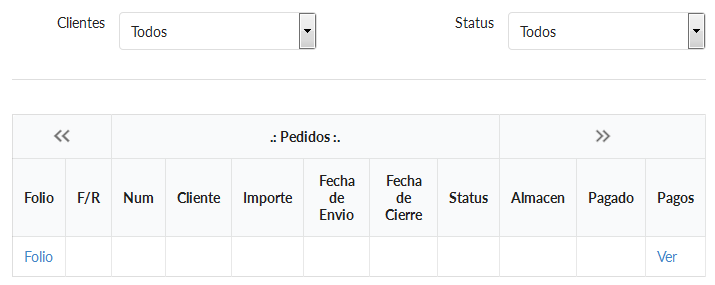
****



Carrito

**Consultar Pedidos:**

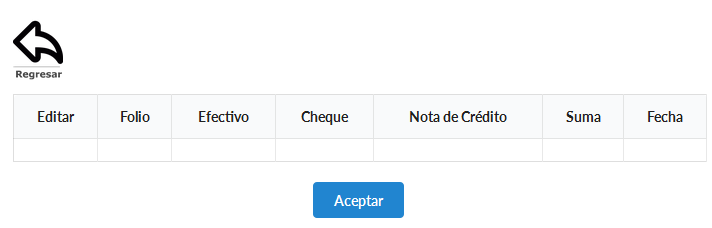
En consultar pedidos muestra todos los pedidos que se han generado, permitiendo mostrar por cliente o por el estatus del pedido entre aprobado o pendiente.



Al seleccionar el folio se muestra el pedido a detalle de todos los productos incluidos en el, cantidades y precios.

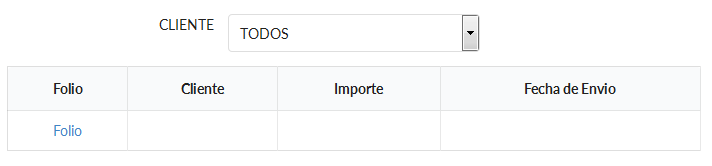


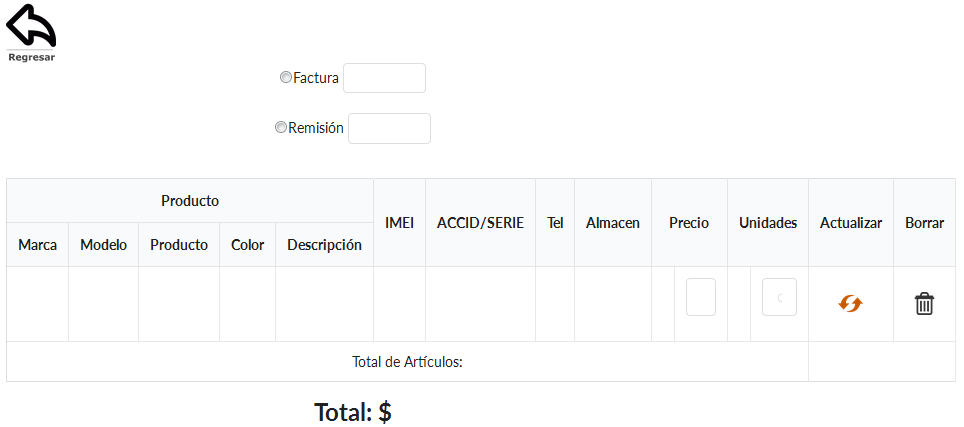
Al seleccionar en pagos muestra la cantidad de pagos realizados para ese pedido, donde se puede seleccionar en editar y permitirá modificar el depósito.



**Aprobar pedidos**

En esta sección muestra a todos los clientes que han hecho un pedido, al seleccionar el folio del pedido muestra en detalle el pedido, donde posteriormente muestra la aprobación del pedido, si es por factura o remisión, la fecha y aprobación del pedido.





# **SELECCIÓN DE COLORES**

