© Universidad Internacional de La Rioja (UNIR)



**Trabajo**

Extracción de las clases de análisis a partir de un documento de información con un enfoque orientado a objetos

**Asignatura**

Metodologías, Desarrollo y Calidad en la Ingeniería de Software

**Profesor**

Omar Uriel Domínguez Mendoza

**Nombre del Alumno:**

Ing. Carlos Pano Hernández

**Fecha de entrega:**

Viernes 4 de Noviembre 2022

Tema 3. Actividades

1

© Universidad Internacional de La Rioja (UNIR)

**Plantamiento del problema:** (Obtención de requisitos)

El equipo de *Azteca Sportswear* desea expandir su E-Commerce: *azteca- sportswear.com* al siguiente nivel:

Se desea crear una tienda que inicialmente contará con un máximo de 10,000 productos y estará conformada por varios vendedores locales. POR AHORA, solo se podrán vender camisetas, sudaderas y chamarras, sin embargo se espera que en un futuro se agreguen muchas mas categorías.

Los requisitos de este Software son los siguientes:

# Back-End:

* **Traducción** al español.
* Existirán varios **usuarios** con las siguientes credenciales/permisos:
  + **Master:** Todos los permisos. Solo existe uno.
  + **Admin:** Podrá aprobar nuevos “salers”. Modificar sus productos y re-categorizarlos.
  + **Saler:** Podrá agregar nuevos productos a su tienda. Crear descuentos. Editarlos
  + **Developer:** Modificaciones generales al estilo de la página. Subir cambios a la misma.
  + **Client:** Acceder a su cuenta, comprar productos.
* El sitio será una aplicación con base en Node.JS-Express.
  + Filtros para los productos.
  + **Templates** para páginas de clientes.
  + Modificar datos.
* Estará desplegada utilizando Oracle Cloud Infrastructure con los siguientes servicios:
  + **Virtual Machine:** Almacenar todos los contenidos.
  + **DevOps:** Pipeline para despliegue.
  + **MySQL Database:** Información del usuario.
  + **Virtual Cloud Network:** Conexión a Internet.
* **Un domino:** *azteca-sportswear.com |* Clientes

Tema 3. Actividades

2

© Universidad Internacional de La Rioja (UNIR)

* **Sub-dominio:** *azteca-sportswear.saler.com |* Panel para vendedores.
* **Sub-dominio:** *azteca-sportswear.admin.com |* Panel para administradores y MASTER.

# Lado del cliente (Front-End):

* El cliente podrá entrar a la página de Internet desde **cualquier browser**

(Chrome, Firefox, Edge y Opera).

* Podrá **iniciar sesión** con su correo y contraseña. No se requiere autenticación de dos factores.
* Agregar al carrito.
  + Pago con PayPal (no se aceptarán datos de tarjetas de crédito).
* Los clientes podrán ver **páginas exclusivas** con **solo los productos** de cada vendedor.
  + Estos productos también podrán ser **filtrados por categoría.**
* Al ingresar a la tienda, verá un Hero Banner en corrusel con las últimas ofertas y códigos de promoción.
  + Seguido de los artículos más vendidos.
  + Cuadros de marcas/vendedores.
  + Todos los productos en bruto con capacidad de filtro.

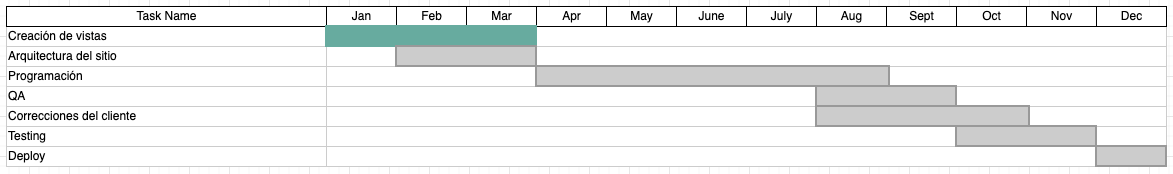
A partir de los requerimientos del equipo de *Azteca Sportswear*, se realizó un estimado del costo total del proyecto y **este fue aceptado.**

Tema 3. Actividades

3

© Universidad Internacional de La Rioja (UNIR)

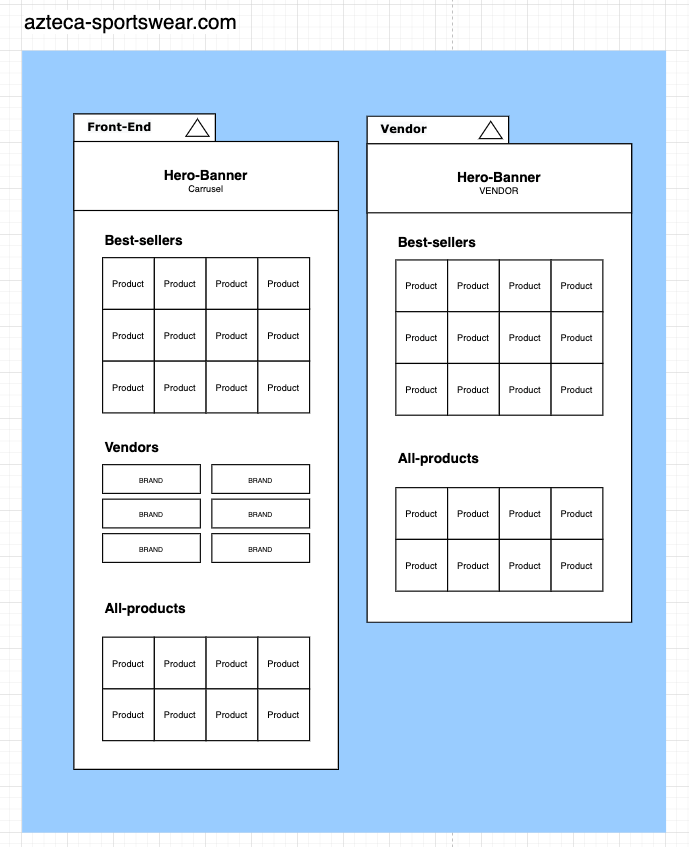
# Desarrollo del Software:

Nos encontramos en la fase de creación de vistas:

Timeline V1

# Wireframes:

El equipo de diseño ha entregado todos los Wireframes necesarios que cumplen con todos los requerimientos de nuestro Front-End:



Wireframe V. 1.0.0.

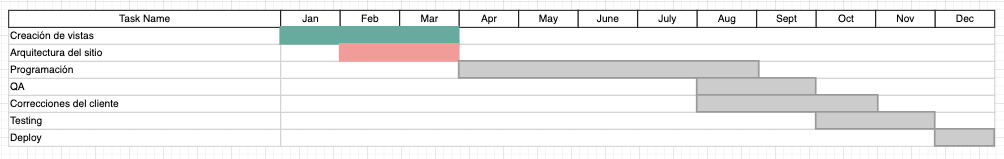
Tema 3. Actividades

4

© Universidad Internacional de La Rioja (UNIR)

Se puede apreciar que el Home de nuestra página contiene el carrusel con el espacio para las promociones necesarias del negocio. Vemos que la **página principal** (index) cuenta con tres secciones distintas: Best-products, vendedores y All products, mientras que nuestra vista de “Vendor” tiene la sección de Best-Sellers y todos los productos. Con este primer avance, se puede proceder a la primera revisión con el cliente, puesto que se utilizarán estas primeras vista para el armado de la arquitectura de nuesto software:

**Primera iteración:** El cliente aprobó nuestras vistas. Los cambios serán realizados por lado del equipo de diseño.



Timeline V2

|  |  |
| --- | --- |
| **Completado** | **Por hacer** |
| Vista Index Vista Vendor | Vista inicio de sesión  Vista visualización/edición de datos Vista de administradores |

Tabla 1. Backlog equipo de diseño

Tema 3. Actividades

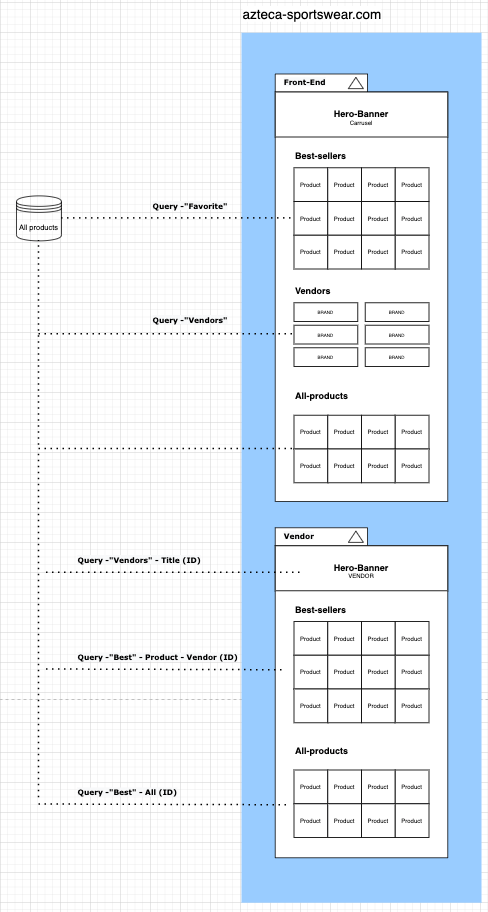
5

© Universidad Internacional de La Rioja (UNIR)

**Arquitectura del sitio:**

Con esta primera entrega, el equipo puede trabajar de manera paralela (ver timeline V2) en la creación de la aplicación para cumplir con las fechas estipuladas del PM.

Veamos ahora un diagrama general de como el Front se alimentará de nuestras bases de datos:



Pre-diagrama funcionamiento Front

Se retoman a continuación los requisitos de nuestro sistema:

|  |  |
| --- | --- |
| **No. de Requisito** | **Descripción** |
| R001 | El sistema debe traducirse automáticamente de  Inglés a Español |
| R002 | Master dará alta/baja a permisos de usuarios  (admin, saler, developer). Clientes podrán crear sus cuentas ellos mismos (permisos básicos) |

Tema 3. Actividades

6

© Universidad Internacional de La Rioja (UNIR)

|  |  |
| --- | --- |
| R003 | Saler podrá agregar productos a su tienda |
| R004 | Saler podrá editar productos y agregar descuentos |
| R005 | Master/admin podrá cambiar los filtros de  producto |
| R006 | Cliente podrá realizar el pago de sus productos |
| R007 | Saler podrá descargar historial de ventas |
| R008 | Admin podrá ver historial de ventas de todos sus  clientes |
| R009 | Usuarios deberán autenticarse |
| R010 | Páginas de productos se crean en automático de  acuerdo a las preferencias |
| R011 | Front se alimenta automáticamente de las bases  de datos (dinámica) |
| R012 | Usuarios tendrán su propio panel de  administración de acuerdo a los permisos |
| R013 | Notificaciones saler: producto creado, nueva  venta |
| R014 | Notificaciones cliente: venta exitosa |
| R015 | Master podrá descargar reportes de ventas y  comisiones de uso de plataforma |
| R016 | Las transacciones sólo podrán ser realizadas con  PayPal |

Tabla 2. Requerimientos

Teniendo presente todos nuestros requisitos, se procede con la descripción de clases del sistema:

|  |  |
| --- | --- |
| **Clase** | **Descripción** |
| Usuario | Consta de la información básica de todo empleado  o cliente. |
| Países | Se encuentran las listas de lugares/regiones para  comprar |

Tema 3. Actividades

7

© Universidad Internacional de La Rioja (UNIR)

|  |  |
| --- | --- |
| Cliente/client | Se agregan las características específicas de un  cliente: cuenta de PayPal, dirección de envío, últimos pedidos, etc. |
| Cart | El cliente podrá visualizar los items agregados a  su carrito de compra |
| PagoCliente | El cliente podrá hacer el pago de su carrito. |
| EncryptPayPal | Esta clase se encargará de la conexión con nuestro  gestor de pagos |
| Master | Conjunto de todos los permisos y capacidad de  agregar nuevos admins o salers, así como modificar información de productos. |
| Saler | Capacidad de agregar productos y descuentos en  sus productos únicos. |
| Developer | Agregar contenido al Front de la página. No  podrán hacer modificaciones en los productos. |
| Products | Constará de toda la información de nuestros  productos. |
| Category | Muestra las categorías compartidas de nuestros  productos |

Tabla 3. Descripción de clases

Una vez teniendo las descripciones de la misma, se realiza un diagrama de clases para tener una vista general del comportamiento del mismo:

Tema 3. Actividades

8

© Universidad Internacional de La Rioja (UNIR)

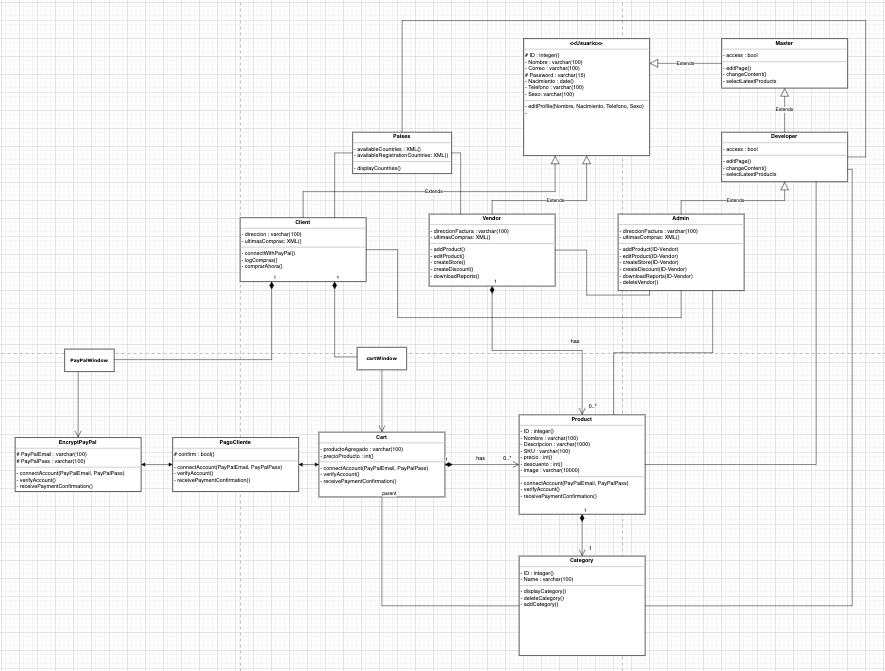


Diagrama 1. Clases

Mismo que se detalla a continuación:

**Clase “usuario”.** Partimos de una clase abstracta que contendrá todos los distintos atributos que todo usuario deberá tener:

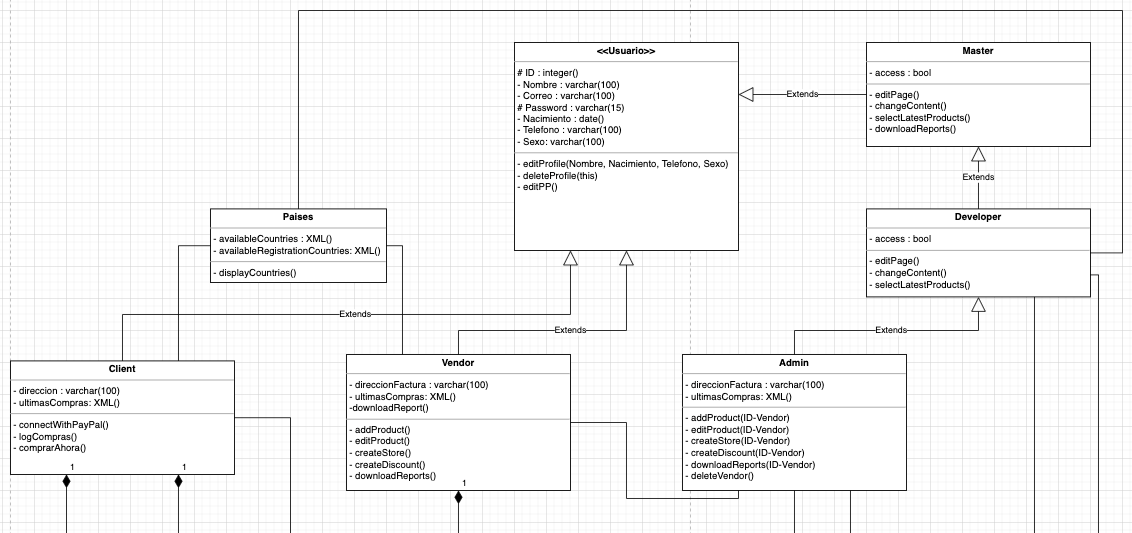


Diagrama 2. Users

Tema 3. Actividades

9

© Universidad Internacional de La Rioja (UNIR)

Podemos ver que de la clase “usuarios” derivan los clientes, vendedores y el Master, un usario padre que podrá tener acceso total. Cada uno con sus distintos comportamientos. Del Master, derivan nuestros desarrolladores y administradores pues queremos que tengan las mismas funciones pero serán restringidas.

Así pues, los clientes podrán hacer edición de sus productos, registrar su cuenta de PayPal y agregar productos a nuestro carrito de compras. Se creó la clases “países” para poder delimitar zonas de compra o bien, envíos.

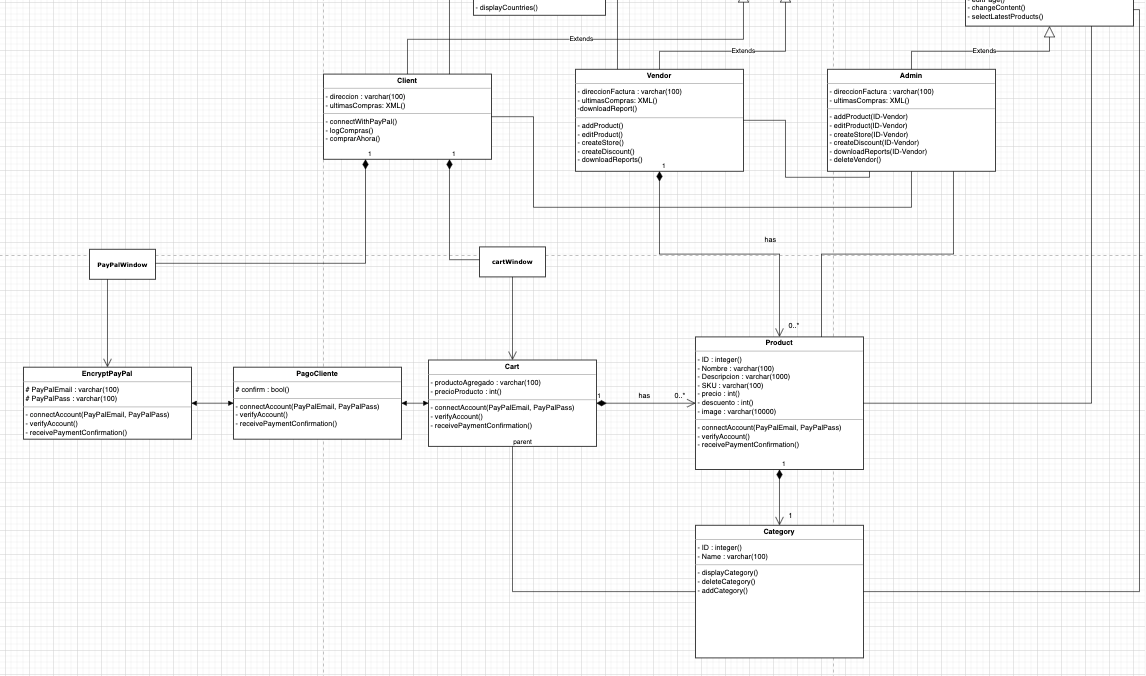
Si seguimos con la siguiente parte del diagrama:

Diagrama 3. Store

Vemos que un vendedor puede tener varios productos, pero si este usuario es elimado, los produtos no pueden existir. Continuando, el carrito de nuesto cliente puede contener varios productos de nuestro vendedor, y es así como el cliente puede comprar varios productos de múltiples salers.

A partir de lo anterior, *Azteca Sportswear* gozará de los siguientes

# beneficios:

* Una cuenta Master que permitirá administrar toda la tienda sin restricciones, así como agregar y eliminar usuarios.

Tema 3. Actividades

10

© Universidad Internacional de La Rioja (UNIR)

* Un perfil de desarrolladores para mantener el código, sin necesidad de ver la información de nuestros clientes.
* Un perfil administrativo para poder dar soporte a clientes.
* Nuestros clientes:
  + Podrán comprar productos de varios vendedores.
  + Podrán agregarlo a un carrito de compras.
  + Podrán hacer el pago correspondiente utilizando PayPal.
  + Así como la edición de todos sus datos.
* Nuestros vendedores:
  + Podrán agregar productos.
  + Editar o borrarlos.
  + Agregar descuentos.
  + Editar su información.

A partir de nuestro diagrama, especificamos nuestras operaciones de cada clase para cumplir con los requerimientos:

|  |  |  |
| --- | --- | --- |
| **Clase** | **Atributos** | **Operaciones** |
| Usuario | Id (identificador único, nombre, correo, password, fecha de nacimiento,  teléfono y sexo) | Editar perfil, borrar perfil, editar imagen, cambiar preferencias, entrar a la tienda |
| Países | Lista de países ahbilitados para vender y comprar | Mostrar lista de países disponibles para el usuario, restringir ventas, aprobar ventas, generar reporte de  ubicación |
| Cliente/client | Del dominio específico (herencia usuario), direccion de envío, registro de  últimas compras | Conectar cuanta PayPal, agregar al carrito, comprar items carrito, descargar historial de compras, editar  compras |

Tema 3. Actividades

11

© Universidad Internacional de La Rioja (UNIR)

|  |  |  |
| --- | --- | --- |
| Cart | Del dominio específico (herencia client), agregar producto, confirmación de producto agregado, solicitar aprobación  de compra | Listar productos agregados, editar productos |
| PagoCliente | Confirmacion de compra | Conexión PayPal, recibir monto de compra, generar recibo, enviar recibo, verificar  últimas credenciales |
| EncryptPayPal | Solicitar correo  usuario, solicitar contraseña usuario | Generar Token de verificación,  recibir pago, autenticar compra, validar compra |
| Master | Del dominio específico (herencia usuario) | Descargar logs de compras, eliminar usuarios, administrar información, editar  información |
| Saler/Vendor | Del dominio específico (herencia usuario), dirección de factura, registro  de últimas compras | Agregar producto, editar productos, crear tienda, crear descuentos, descargar reportes |
| Developer | Del dominio específico (herencia  usuario) | Editar categorías, cambios contenido, actualizar  productos estrella |
| Products | Id único de identificación, nombre, descripción, SKU, precio,  descuento, imagen | Mostrar precio, calcular descuento, mostrar variaciones |

Tema 3. Actividades

12

© Universidad Internacional de La Rioja (UNIR)

|  |  |  |
| --- | --- | --- |
| Category | Id, listado de clases | Mostrar categorías, borrar  categorías, agregar categorías |

Tabla 4. Atributos y operaciones

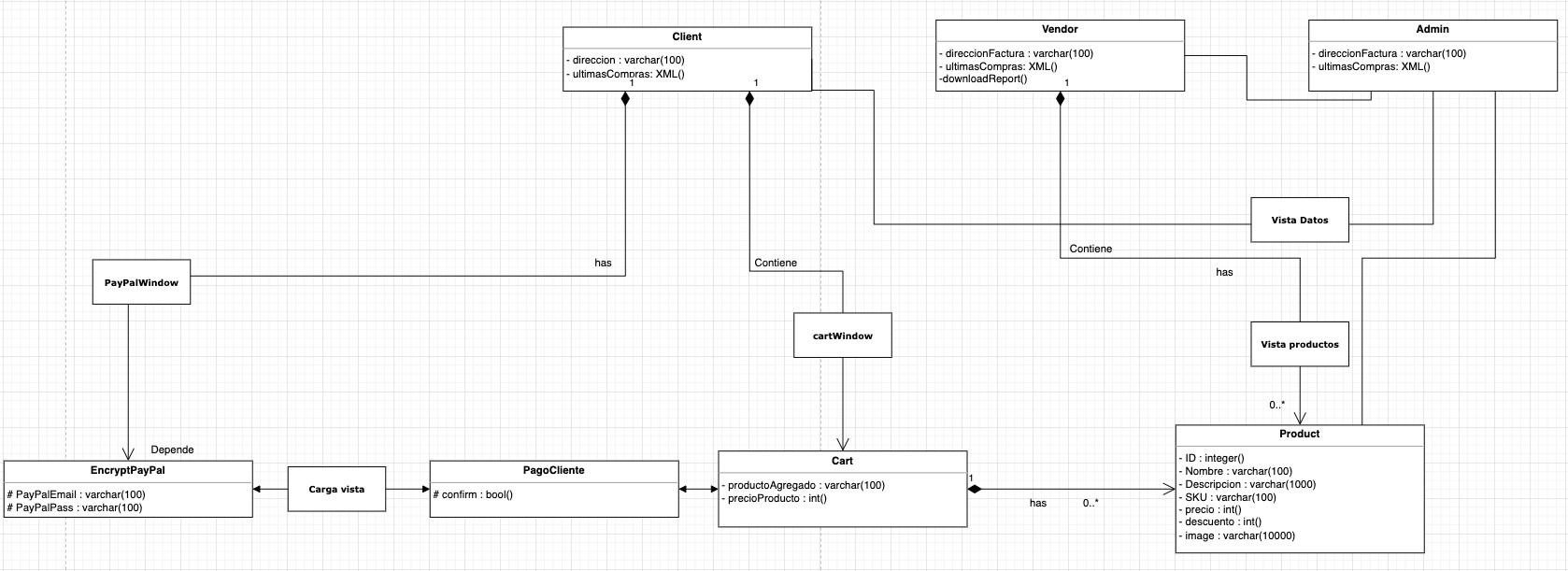
Lo anterior, nos permite definir nuestro diagrama de clases y relaciones:

Diagrama 3.5. Clases y Relaciones

Una vez vaciadas todas las clases, procedemos con la creación de un diagrama de Casos de Uso para una representación de los métodos principales del sistema:

Tema 3. Actividades

13

© Universidad Internacional de La Rioja (UNIR)

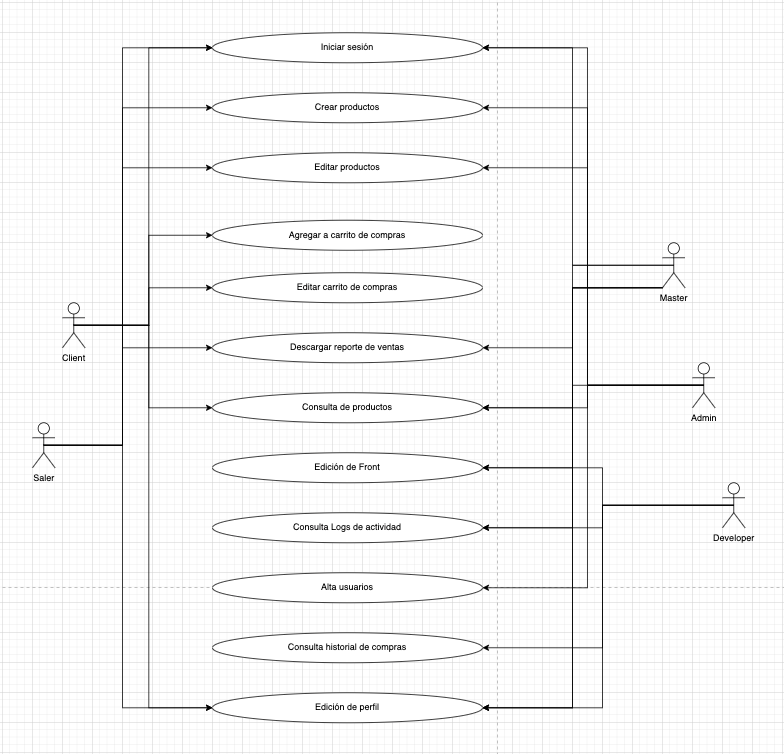


Diagrama 4. Casos de Uso

An el anterior se puede ver de manera gráfica como algunos de los perfiles comparten ciertas capacidades del sistema, la cual ayudará a la simplificación de código.

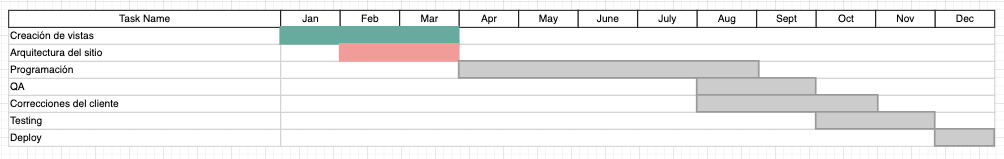
# Modelo de desarrollo:

A partir de los diagramas anteriores y la naturaleza del proyecto, se escogió en esta primera etapa un modelo en cascada. Se conoce que no es lo óptimo, sin embargo, con base en el “Timeline V2”:

Tema 3. Actividades

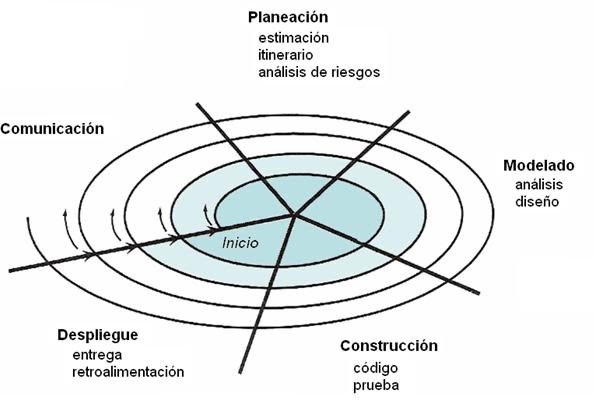
14

© Universidad Internacional de La Rioja (UNIR)



Timeline V2

Vemos que existen algunos procesos paralelos para una entrega limpia. En esta misma, se tienen consideradas las fases de QA + las correcciones del cliente. A partir de esta primera entrega con todos los requisitos, se recomienda aplicar un modelo evolutivo para la creación de nuevas características al sistema:



Se enlistan algunas mejoras con el modelo evolutivo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Planeación** | **Modelado** | **Construcción** | **Despliegue** | **Comunicación** |
| Implementar | Uso de | Creación de | Testing | Verificación con |
| autenticación | librearías | middleware |  | cliente |
| de doble | externas | para autenticar |  |  |
| factor en |  | visibilidad en la |  |  |
| inicio de |  | siguiente vista. |  |  |
| sesión |  |  |  |  |

Tema 3. Actividades

15

© Universidad Internacional de La Rioja (UNIR)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QA** doble | Diseño de | Implementación | Evaluación | Recepción de |
| factor. | nuevas | de diseño | de nuevos | nuevo diseño |
| Nuevos | plantillas |  | diseños |  |
| templates | para |  |  |  |
| para “salers” | nuestros |  |  |  |
|  | vendedores |  |  |  |

Tabla 5. Implementación de modelo evolutivo

# Stakeholders

Se determinan los principales interesados en la realización de este proyecto:

1. Director financiero *Azteca Sportswear*
2. Proyect Manager
3. Clientes externos interesados en comprar marcas locales
4. Desarrolladores
5. Administradores de tienda

# Conclusión:

A partir de lo anterior, vemos la importancia de comenzar un proyecto con fuertes bases. Con base en mi experiencia, he notado que una fuerte relación con el equipo de diseño ayuda a reducir el procentaje de errores al ya tener un comportamiento/visualización gráfica. Dicho lo anterior, se deicidió a realizar un pequeño Framework del sistema para generar una idea más clara del proyecto y así comenzar con el modelado de clases y su respectiva explicación. Esto facilitó el proceso para la generación de los casos de usos y demás.

Se notó de igual manera que siguiendo un proceso en cascada puede agilizar la primera entrega del proyecto, siempre y cuando evolucione a un modelo más complejo. De esta manera podemos ralizar trabajos en forma paralela y reducir costos para la empresa.

# Bibliografía:

* Booch, G., Rumbaugh, J. y Jacobson, I. (2005). Unified Modeling Language User Guide (2° ed.). Addison Wesley.

Tema 3. Actividades

16

© Universidad Internacional de La Rioja (UNIR)

* Hanssen, G. K., Westerheim, H. y Bjerson, F. O. (2005). Tailoring RUP to a defined project type: A case study. Actas de 6th International Conference on Product Focused Software Process Improvement, PROFES, Oulu, Finland. Springer-Verlag **3547, 314-327.**
* IEEE. (1998). IEEE 830-1998 - IEEE Recommended practice for software requirements specifications [Standard]. The Institute of Electrical and Electronics Engineers.https://standards.ieee.org/standard/830-1998.html
* Jacobson, I., Booch, G. y Rumbaugh, J. (2000). **El proceso unificado de desarrollo de software.** Addison Wesley.
* Kruchten, P. (2003). The Rational Unified Process: An Introduction (3° ed.). Addison-Wesley.

Tema 3. Actividades

17