NOVIEMBRE, 2016



**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**ASIGNATURA:**

PROYECTO INTEGRADOR DE INGENIERÍA DE SOFTWARE

11:00-12:00 P.M.

**MAESTRA:**

ARCELIA JUDITH BUSTILLOS MARTINEZ

**ALUMNOS:**

RAMIREZ SALAZAR LUIS FERNANDO

JOSÉ MARIO SANZ LÓPEZ.

**ARTEFACTOS DE LA FASE DE ELABORACIÓN**

**TABLA DE CONTENIDO**

1. MODELO DEL DOMINIO
   1. Introducción
   2. Diagrama de secuencia del sistema
      1. Añadir artículos a la BD
      2. Añadir artículos al carrito
   3. Visualización de conceptos
   4. Identificación de clases conceptuales usando la técnica de análisis lingüístico
      1. Escenario principal de éxito.
   5. Categorías de clases conceptuales
   6. Clases conceptuales candidatas
   7. Asociaciones
      1. Localizaciones de asociaciones
      2. Asociaciones en detalle (Multiplicidad)
   8. Atributos
   9. Modelo conceptual final
   10. Contratos
2. MODELO DE DISEÑO
   1. Análisis de requisitos del sistema
   2. Análisis de requisitos software
   3. Diseño preliminar
   4. Diseño detallado
   5. Codificación y pruebas
   6. Exploración y mantenimiento
3. DOCUMENTO DE LA ARQUITECTURA DEL SOFTWARE
   1. Lista de interesados y sus roles
      1. Lista de interesados
      2. Roles de los interesados
   2. Diagrama de contexto
   3. Arquetipos y sus relaciones
      1. Arquetipos
      2. Relación entre arquetipos
   4. Clasificación y priorización de los requisitos funcionales y no funcionales
      1. Requisitos funcionales
      2. Requisitos no funcionales
   5. Decisiones de la arquitectura
      1. Parte funcional del sistema
      2. Parte no funcional del sistema
   6. Escenarios del sistema
      1. Casos de uso
      2. Matriz de requisitos funcionales
   7. Diagramas de casos de uso seleccionados
   8. Modelo inicial y Estructura del sistema
      1. Ventas
         1. Modelo inicial
         2. Atributos

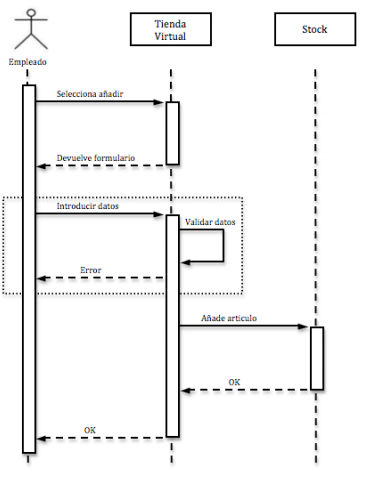
1. MODELO DE DATOS
2. MODELO DE PRUEBAS
   1. Modelo en V
   2. Formato de pruebas
3. MODELO DE IMPLEMETACIÓN
   1. Diagrama de componentes
   2. Diagrama de despliegue
   3. Modelo de pruebas
      1. Tipos de pruebas
4. GUIONES DE CASOS DE USO, PROTOTIPOS DE IU
   1. Guiones de casos de uso
   2. Prototipos de IU
5. **MODELO DEL DOMINIO**
   1. INTRODUCCION

El modelo del dominio muestra clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos.

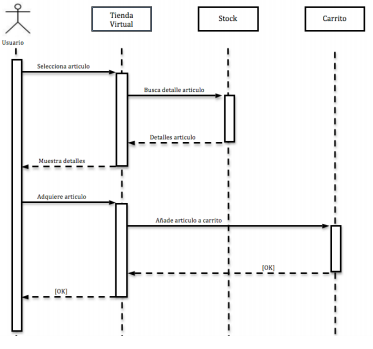
La etapa orientada a objeto esencial del análisis es la descomposición de un dominio de interés en clases conceptuales individuales u objetos

Los modelos de dominio pueden mostrar:

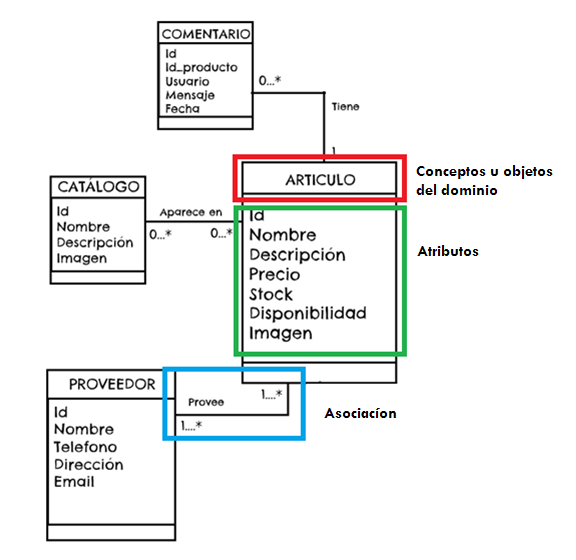
* Objetos del dominio o clases conceptuales
* Asociaciones entre las clases conceptuales
* Atributos de las clases conceptuales
  1. DIAGRAMA DE SECUENCIA DEL SISTEMA
     1. AÑADIR ARTICULOS A LA BD

****

* + 1. AÑADIR ARTICULOS AL CARRITO

****

* 1. VISUALIZACIÓN DE CONCEPTOS



* 1. IDENTIFICACIÓN DE CLASES CONCEPTUALES USANDO LA TÉCNICA DE ANÁLISIS LINGÜÍSTICO
     1. ESCENARIO PRINCIPAL DE ÉXITO

1.- El usuario registrado llena su carrito de artículos de su agrado.

2.- El usuario comienza una nueva compra en la tienda.

3.- El usuario introduce su forma de pago

4.- El sistema registra la venta en línea y presenta la descripción de cada artículo, con el precio de ellos, las especificaciones y la suma parcial.

5.- El usuario repite los pasos 3, las veces que lo requiera.

6.- El sistema presenta el precio total de todos los artículos seleccionados.

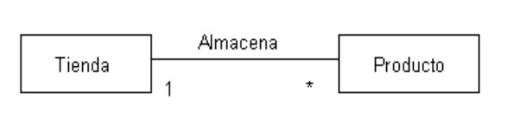
7.- El administrador constantemente introduce nuevos artículos a la BD

* 1. CATEGORÍAS DE CLASES CONCEPTUALES

|  |  |
| --- | --- |
| CATEGORÍA | EJEMPLOS |
| Objetos tangibles o físicos | Registro |
| Especificaciones, diseño o descripciones de las cosas | Especificación del producto, especificación de categoría |
| Lugares | Tienda virtual |
| Transacciones | Venta, pago |
| Líneas de la transacción | Línea de venta |
| Roles de la gente | Administrador |
| Contenedores de otras cosas | Tienda virtual |
| Cosas en un contenedor | Artículos, comentarios |
| Otros sistemas informáticos o electromecánicos externos al sistema | Sistema de autorización de pago por tarjetas |
| Conceptos abstractos | Ansia |
| Organizaciones | Departamento de ventas |
| hechos | Venta, pago, llegada de artículos |
| Procesos (normalmente no se presentan como conceptos) | Venta de un producto |
| Reglas y políticas | Política de cancelación |
| Catálogos | Catálogo de productos |
| Registro de finanzas, trabajo, contratos, cuestiones legales | Recibo, libro mayor |
| Instrumentos y servicios financieros | Stock |
| Manuales, documentos, artículos de referencia, libros. | Lista de cambios de precios diarios. |

* 1. CLASES CONCEPTUALES CANDIDATAS
* Tienda
* Articulo
* Venta
* Pago
* Catálogo
* Registro
* Línea de venta
* Especificación del producto
  1. ASOCIACIONES

Es una relación entre tipos que indica alguna conexión significativa e interesante.



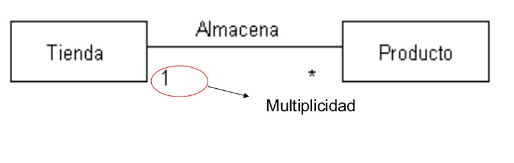
* + 1. LOCALIZACIONES DE LAS ASOCIACIONES

|  |  |
| --- | --- |
| CATEGORÍAS | EJEMPLOS |
| A es una parte física de B |  |
| A es una parte lógica de B | Venta de producto – venta |
| A esta físicamente contenido en B | Tienda – producto |
| A esta contenido lógicamente en B | Descripción del producto – catalogo |
| A es una descripción de B | Descripción del producto - producto |
| A es un elemento de línea en una transacción o reporte B | Ventas de productos -venta |
| A se conoce en B | Venta-tienda |
| A es miembro de B | Administrador-tienda |
| A es una unidad organizacional de B | Catalogo-tienda |
| A usa o dirige a B | Administrador – tienda |
| A se comunica con B | Cliente – tienda |
| A se relaciona con una transacción de B | Pago – venta |
| A es una transacción relacionada con otra transacción de B | Pago – venta |
| A es propiedad de B | Venta – tienda |

* + 1. ASOCIACIONES EN DETALLE (MULTIPLICIDAD)

Multiplicidad:

* \*.- 0 o mas
* 1..- uno o mas
* 5.- exactamente 5
* 3, 5, 8.- exactamente 3, 5, 8

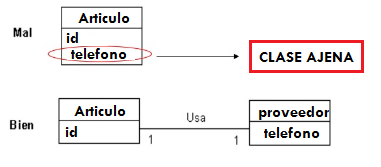


* 1. ATRIBUTOS

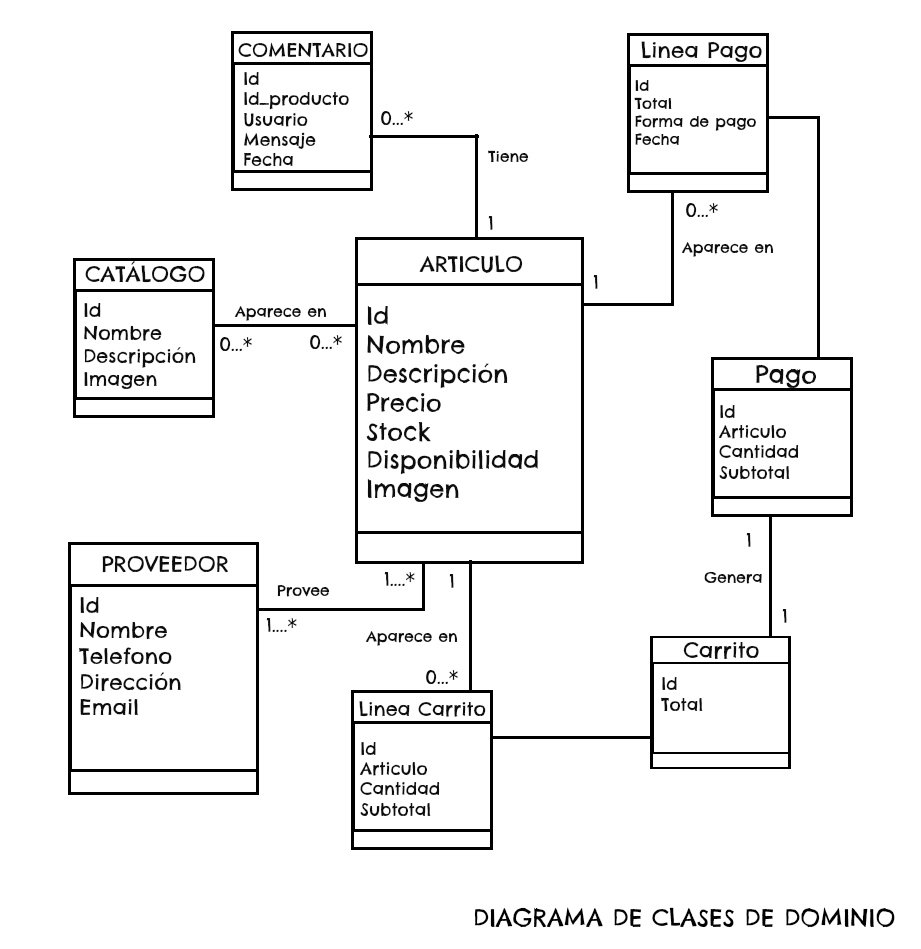
Atributo es el valor de datos lógico de un objeto.

Los atributos en un modelo de dominio deberían ser preferiblemente atributos simples o tipos de datos.

Se recomienda que se relacionen las clases conceptuales por asociaciones y no por atributos.



* 1. MODELO CONCEPTUAL FINAL



1.10 CONTRATOS

**Contrato C01: LoginUsuario (Login de usuario).**

* Operación: LoginUsuario()
* Referencia Cruzada: Caso de uso: iniciar sesión.
* Pre-condiciones: Hubo un Previo registro.
* Pos-condiciones:
  + se creó una instancia de usuario

**Contrato C02: AgregarCarrito (Agregar artículos al carrito)**

* Operación: AgregarCarrito(Producto)
* Referencia Cruzada: Caso de uso: Agrega artículos al carrito.
* Pre-condiciones: Usuario logueado.
* Pos-condiciones:
  + se cargaron atributos de productos
  + Se calculó atributo de producto(modificación de atributos)

**Contrato C03: crearNuevaVenta**

* Operación: crearNuevaVenta ()
* Referencia cruzada: caso de uso: Confirmar venta
* Pre-condiciones: ninguna.
* Pos-condiciones:
  + Se creó una instancia de Venta v (creación de instancias)
  + V se asoció con el registro (formación de asociaciones)
  + Se inicializaron los atributos de v

**Contrato C04: IntroducirArticulo**

* Operación: IntroducirArticulo (articuloID: articuloID, cantidad:integer)
* Referencia Cruzada: Caso de uso: Confirmar venta
* Pre-condiciones: Hay una venta en curso.
* Pos-condiciones:
  + Se creó una instancia de LineaDeVenta ldv (creación de instancias)
  + Ldv Se asoció con la venta actual (formación de asociaciones)
  + Ldv.cantidad paso a ser cantidad (modificación de atributos)
  + Idv se asoció con una EspecificacionDelProducto, en base a la coincidencia del articuloID (formación de asociaciones)

**Contrato C05: AltaCategoria**

* Operación: AltaCategoria
* Referencia Cruzada: Caso de uso: Alta categoría
* Pre-condiciones: tipo de usuario: administrador
* Pos-condiciones:
  + Se creó una instancia de categoría
  + Se cargaron los atributos de productos
  + Se calculó atributo de producto(modificación de atributos)

**Contrato C06: Cerrar sesión**

* Operación: CerrarSesion()
* Referencia Cruzada: Caso de uso: Iniciar sesión, Cerrar Sesión.
* Pre-condiciones: Hay un usuario logueado.
* Pos-condiciones:
  + se eliminó instancia de usuario (destrucción de instancia).

1. **MODELO DE DISEÑO**
2. **DOCUMENTO DE LA ARQUITECTURA DEL SOFTWARE**
   1. LISTA DE INTERESADOS Y SUS ROLES
      1. LISTA DE INTERESADOS

* Administradores
* Clientes
* Proveedores
  + 1. ROLES DE LOS INTERESADOS
* Administrador:

Tiene el acceso a la mayoría de áreas existentes en el sistema con la finalidad de llevar un control de todo lo que se maneja con respecto a compra y venta del producto del negocio.

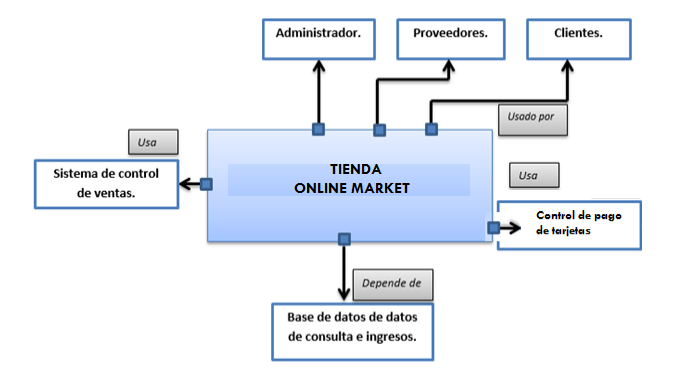
* Clientes:

Este interesado toma más importancia cuando realiza una compra.

* Proveedores:

Personal externo encargado de suministrar el producto en la empresa, es el responsable de las entradas del producto dentro de la empresa.

* 1. DIAGRAMA DE CONTEXTO



* 1. ARQUETIPOS Y SUS RELACIONES
     1. ARQUETIPOS
* Persona:

Este arquetipo se representa a los involucrados del proyecto, ósea, los diferentes tipos de usuarios que tiene nuestro proyecto.

Son la herramienta muy útil en el diseño de interfaces por que le dan una cara humana a la creación de la experiencia del usuario y sirven para la guía del diseño.

* Producto:

Este arquetipo se representa los productos que se le ofrecen al cliente. Abarcan comportamiento desde la localización en la tienda, análisis del producto, etc.

* + 1. RELACIÓN ENTRE ARQUETIPOS
* Persona-Producto:

La relación entre estos dos arquetipos se da en primera manera, en que el arquetipo persona puede hacerse de un producto o servicio.

Otra relación es que un producto, puede ser dado de alta por una persona, en cargada del sistema, ya sea el administrador, o bien que el encargado de comprar más productos para tenerlos en el stock de la tienda.

* 1. CLASIFICACIÓN Y PRIORIZACIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES
     1. REQUISITOS FUNCIONALES

|  |  |
| --- | --- |
| REQUISITOS FUNCIONALES | PRIORIDAD |
| 1.- El administrador deberá tener la posibilidad de buscar en base de datos los productos que se le piden por parte del cliente | Alta |
| 2.- El sistema deberá proporcionar visores adecuados para que el encargado de la venta pueda ver los productos que hay existentes y cuáles son los que puede comprar. | Alta |
| 3.- El sistema deberá de generar el costo total de la venta incluyendo el IVA. | Alta |
| 4.- El sistema deberá de generar un recibo que indique los productos que se compraron | Alta |
| 5.- El sistema debe de tener una función de autorización de pago usando tarjetas de crédito | Media |
| 6.- El sistema debe de considerar realizar descuentos especiales para clientes frecuentes, tomando el monto acumulado de ventas | Media |

* + 1. REQUISITOS NO FUNCIONALES

|  |  |
| --- | --- |
| REQUISITOS NO FUNCIONALES | PRIORIDAD |
| 1.- Acceso seguro y validación de usuarios | Alta |
| 2.- interfaz amigable | Media |
| 3.- Respuestas rápidas | Alta |
| 4.- Operación de pago confiable y seguro | Alta |
| 5.- Tolerancia a fallos inesperados | Media |
| 6.- Adaptabilidad, seguridad, estabilidad, portabilidad, etc. | Media |

* 1. DECISIONES DE LA ARQUITECTURA

En este punto se especifican las decisiones que se tomaran en cuenta para posteriormente elegir un tipo de arquitectura de software que se adecue al proyecto. Es importante aclarar que las decisiones de la arquitectura están basadas en los requisitos del sistema, hablando de los funcionales tanto como de los no funcionales o requisitos de calidad, ya que estos nos indican como el sistema en sí, estará distribuido en módulos o de que partes estará compuesto, y nos ayuda a establecer arquetipos para visualizar el sistema de una forma más general y consistente.

* + 1. PARTE FUNCIONAL DEL SISTEMA
* Modelo arquitectónico cliente servidor:

Este modelo arquitectónico cuenta con las características de mantener una transferencia de información de manera distribuida entre los distintitos módulos que conforman el sistema en base a peticiones y respuestas por medio del cliente (usuario) y el servidor quien mantiene toda la información importante en tiempo real para ser consultada. Se tomó la decisión de considerar este modelo ya que cumple con las necesidades del proyecto en aspecto de consulta y almacenamiento de información.

* Estilo de servidores y servicios compartidos:

Es un modelo de sistema distribuido que muestra como el procesamiento y los datos pueden distribuirse en una serie de componentes, conformados por una serie de servidores que brindan un determinado servicio (impresión, datos, email, etc.). Serie de clientes que utilizan esos servidores, y la red que permite conectar ambas partes.

* + 1. PARTE NO FUNCIONAL DEL SISTEMA
* Rendimiento:

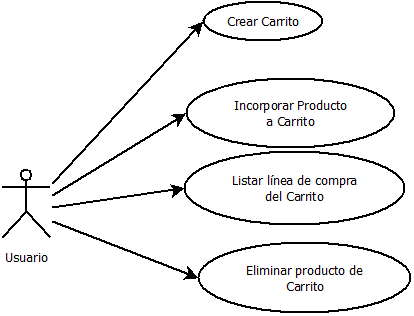
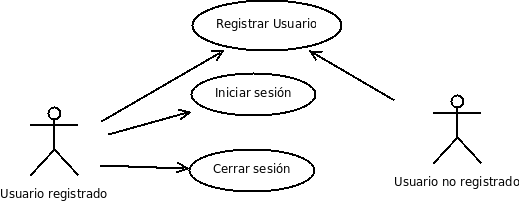
Se toma como decisión rendimiento como una característica con la cual debe portar el sistema con la finalidad de que soporte múltiples peticiones por los usuarios en los distintos módulos y no se presenten fallos al momento de solicitar la información.

* Disponibilidad:

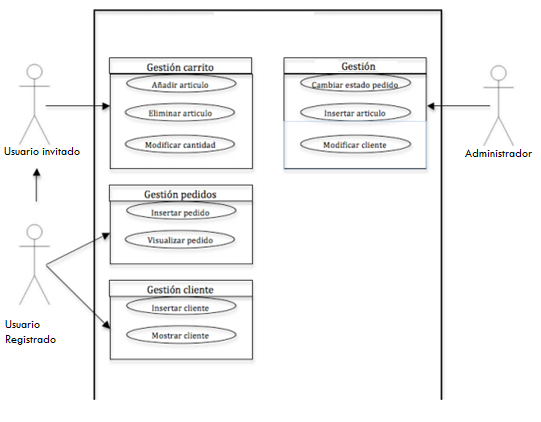
La métrica empleada para medir la disponibilidad es el porcentaje de tiempo que un sistema es capaz de realizar las funciones para las que está diseñado. En lo que se refiere a los sistemas de ventas, la disponibilidad es el porcentaje de tiempo que el servicio del sistema está activo y en funcionamiento.

* 1. ESCENARIOS DEL SISTEMA
     1. CASOS DE USO

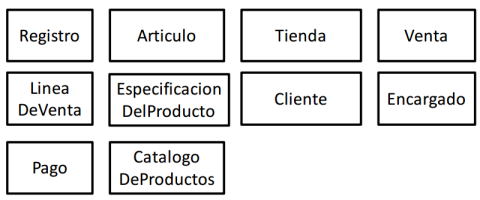


* 1. DIAGRAMAS DE CASOS DE USO SELECCIONADOS



* 1. MODELO INICIAL Y ESTRUCTURA DEL SISTEMA
     1. VENTAS
        1. MODELO INICIAL

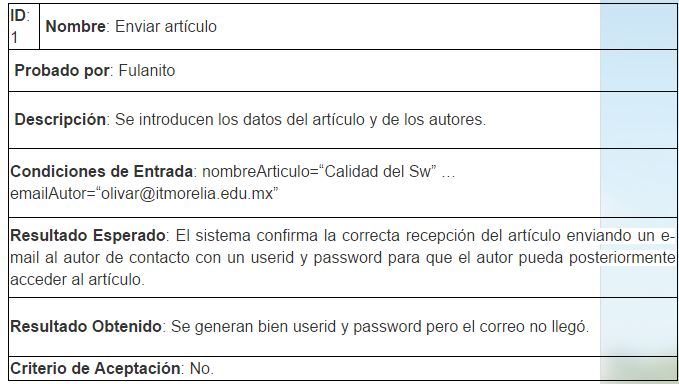


* + - 1. ATRIBUTOS

|  |  |
| --- | --- |
| CONCEPTO | ATRIBUTO |
| PAGO | cantidad: Se debe capturar una cantidad para determinar si se proporciona el pago suficiente y calcular el cambio |
| ESPECIFICACION DEL PRODUCTO | Descripción: Para mostrar la descripción en una pantalla o recibo.  Precio: Para calcular el total de la venta y mostrar el precio de la línea de venta. |
| VENTA | Fecha: Un recibo es un informe en papel de una venta. Normalmente muestra la fecha y la hora de la venta. |
| LINEA DE VENTA | Cantidad: Para registrar la cantidad introducida, cuando hay más de un artículo en una línea de venta. |
| TIENDA | Dirección: El recibo requiere el nombre y la dirección de la tienda. |

1. **MODELO DE DATOS**
2. **MODELO DE PRUEBAS**

**FORMATO PARA EL PLAN DE PRUEBAS**



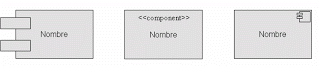
1. **MODELO DE IMPLEMETACIÓN**
   1. DIAGRAMA DE COMPONENTES

Un componente es una parte física de un sistema (modulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con atributos y métodos pueden ser implementados en los componentes.

Los componentes pueden ser:

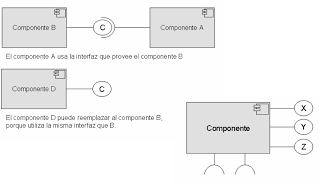
* Archivos
* Código fuente + Cabeceras
* Librerías compartidas (DLLs)
* Ejecutables
* Paquetes

Representación simple de un Componente



¿Por qué utilizar un Diagrama de Componentes?

1. Nos permite ver el modelado de un sistema o subsistema
2. permite especificar un componente con interfaces bien definidas.



* 1. DIAGRAMA DE DESPLIEGUE

El Diagrama de Despliegue es un diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Usos que se les da a los diagramas de despliegue son para modelar:

* Sistemas cliente-servidor
* Sistemas completamente distribuidos

El elemento principal del diagrama son los NODOS.

 Los nodos representan un recurso físico:

* Computadoras
* Impresoras
* Servidores
* Dispositivos externos

 Los nodos pueden ser interconectados mediante líneas para describir una estructura de red.

* 1. MODELOS DE PRUEBAS
     1. TIPOS DE PRUEBAS
* Pruebas de Aceptación o Casos de Uso
* Pruebas de Integración/Sistema o Diagramas de Secuencia/Escenarios
* Pruebas Unitarias o Clases/Módulos
* Pruebas de Regresión
* Pruebas de Facilidad de Uso
* Pruebas de Cobertura
* Pruebas de Rendimiento.

1. **GUIONES DE CASOS DE USO, PROTOTIPOS DE IU**
   1. GUIONES DE CASOS DE USO
2. **ALTA DE USUARIO:**

Resumen de la funcionalidad:  
Da de alta un usuario en la base de datos, se mira que no exista en la BD, introducir los datos del usuario.

Actor: Usuario invitado  
Casos de usos relacionados: Registrarse, consulta usuario.  
Precondición: El usuario no puede existir en la base de datos.

Función principal:

* El usuario escoge la opción de Alta usuario
* El sistema le presenta un formulario a rellenar por el usuario donde le pide: Login, password, repetir password, nombre, apellidos, dirección, código postal, población, provincia, teléfono, email, cuenta, etc.

1. **BAJA DE USUARIO:**

Resumen de la funcionalidad:  
Modifica los datos del usuario registrados en la base de datos.

Actor: Usuario registrado   
Casos de usos relacionados: Baja usuario, Alta usuario, Modifica usuario, Registrarse.  
Precondición: El usuario debe de existir en la base de datos

Función principal:

* El usuario pulsa la opción consulta datos.
* El sistema mostrara los datos del usuario.

1. **MODIFICA EL USUARIO:**

Resumen de la funcionalidad:   
Modifica los datos del usuario registrados en la base de datos.

Actor: Usuario identificado  
Casos de usos relacionados: Consulta usuario.  
Precondición: El usuario se ha de registrar y existe en la base de datos.

Función principal:

* El usuario está consultando sus datos, edita los campos del formulario deseado, y pulsa la opción modificar datos.
* El sistema grabara los datos del usuario.

**EN EL REFINAMIENTO DE LA ETAPA DE CONSTRUCCIÓN SE REFINARAN LOS DEMÁS CASOS DE USO.**

* 1. PROTOTIPOS DE IU.

La interfaz queda identificada de la siguiente forma:

* En la claridad de objetivos Los objetivos quedan definidos en tres bloques:

El frontal superior donde aparece el menú, el bloque vertical derecho donde aparecen los temas disponibles a seleccionar por el usuario y la zona central donde aparecerán los contenidos que se hayan solicitado desde el menú o bloque derecho.

* En su visibilidad:

La interfaz se identifica claramente por sus tres bloques indicados anteriormente, bloque de temas, bloque de menús o acciones y bloque de contenidos.

* En su adecuación a los usuarios:

Los enlaces son textuales, los colores y tamaños de las letras son uniformes y comunes en todo el conjunto de la interfaz.

* En la consistencia y estándares:

Los menús situados en la parte superior e izquierda están en todas las páginas para facilitar la navegación por la web.

* En su flexibilidad y eficacia de uso:

El usuario puede acceder a todas las acciones y contenidos de manera sencilla y sin repetir pasos.

* En el diseño minimalista:

No existe ninguna página que cargue imágenes u otros contenidos que dificulten el acceso rápido a la aplicación.

