Andres Malvestiti

andres-malvestiti@hotmail.com – Legajo: 41946

Profesor

Mario Bressano

Encom

Final JAVA 2017



Índice

[Introducción 2](#_Toc499766018)

[Definición del Sitio Web 4](#_Toc499766019)

[Objetivos del sitio 4](#_Toc499766020)

[Descripción del sitio 4](#_Toc499766021)

[Definición de la Audiencia 4](#_Toc499766022)

[Definición de los contenidos del sitio 5](#_Toc499766023)

[Agrupación de los contenidos 5](#_Toc499766024)

[Identificar los requerimientos funcionales 5](#_Toc499766025)

[Definición de la estructura del sitio 6](#_Toc499766026)

[Mapa del sitio 6](#_Toc499766027)

[Definicion de los limites del sistema 8](#_Toc499766028)

[Requerimientos generales 8](#_Toc499766029)

[Estructura del Sistema 9](#_Toc499766030)

[Diagrama de bloque 9](#_Toc499766031)

[Base de datos 10](#_Toc499766032)

[Diagrama entidad-Relación 10](#_Toc499766033)

[Capa Dao 11](#_Toc499766034)

[Diagramas de Diseño 14](#_Toc499766035)

[CRUD Productos 14](#_Toc499766036)

[diagrama de herencia 14](#_Toc499766037)

[código fuente 14](#_Toc499766038)

[Annotations y xml 14](#_Toc499766039)

[http error pages 14](#_Toc499766040)

[endpoints 16](#_Toc499766041)

[empaquetamiento y despliegue 16](#_Toc499766042)

[Conclusión 16](#_Toc499766043)

# Introducción

En el presente informe se especifica la temática seleccionada que da motivo a la creación de un Sistema Web implementado en JAVA. Definición de la audiencia, temática, funcionalidades genéricas, estructura del sitio y ciertos aspectos de accesibilidad.

Posteriormente se dispone de la estructura del Sistema en diagramas de bloque, enumerando cada una de las capas abstractas que la componen.

Se establece la documentación del diseño como Diagramas de Secuencia de Sistema (DSS), Diagrama de Clases de Diseño (DCD), Diagrama de Secuencia de Diseño (DSD) y Diagrama de Herencia JAVA. Además del Diagrama Entidad-Relación de la Base de Datos seleccionada (MySql en un principio).

Uno de los objetivos principales del desarrollo del Sistema es la robustez, lo que da pie a la aplicación de patrones de diseño y estrategias descriptas posteriormente en este informe.

Finalmente se detalla el release y el host donde se encuentra ubicado el sitio.

El Sitio Web se realiza actualmente con fines académicos y está disponible en la web, no obstante, a futuro será modificado para seguir una estructura API REST contando con la incorporación de frameworks como Spring y Hibernate.

|  |
| --- |
| Encom |
| Sitio Web |
| Primera Parte |

# Definición del Sitio Web

## Objetivos del sitio

Un sitio web capaz de controlar un negocio de ventas de insumos de computación mayorista, que cumplirá con los siguientes requerimientos:

* Login e identificación de distintos tipos de usuarios (administrador, cliente).
* CRUD de productos a comercializar.
* CRUD de usuarios(clientes).
* Listar los productos y mostrar distinta información según el tipo de usuario (invitado, público o administrador).
* Formulario de contacto.
* Gestión de carritos de compra y checkout (fuera de alcance).

## Descripción del sitio

Encom es un sitio web denominado e-commerce ya que representa una tienda online, permite comprar distintos productos tecnológicos, obtener cotizaciones, etc.

Dispone de tres distintos tipos de vistas, que las mismas a su vez dependen del tipo de “identificación” o usuario que este viendo la página, estas son:

* Vista del administrador: esta vista permite agregar, modificar y eliminar los distintos productos que ofrece la tienda en la web (ABM de los productos disponibles). A su vez permite llevar un tipo de control sobre los usuarios registrados.
* Vista del usuario registrado: permite agregar productos al carrito de compras y enviar formularios de contacto.
* Vista del guest o usuario sin registrar: permite visualizar la lista de productos y enviar formulario de contacto.

# Definición de la Audiencia

Encom está destinada a una audiencia específica, con cierto grado de conocimiento en el área de las tecnologías y la computación. Por ende, un cierto conocimiento, básico al menos, en la navegación Web y los tipos de productos que se encuentran disponibles a la venta.

# Definición de los contenidos del sitio

## Agrupación de los contenidos

* Categorías: Los productos disponibles se encuentran categorizados según sus características y rubro al cual pertenecen.
  + Almacenamiento.
    - HDD.
    - SSD.
  + Memorias.
    - DDR3.
    - DDR4.
  + Fuentes.
    - Genéricas.
    - 80+.
  + Motherboards.
    - AM4.
    - 1150.
  + Procesadores.
    - AMD.
    - Intel.
* About: Se detallan los distintos medios para que los visitantes se comuniquen con el negocio.
  + Formulario de contacto.
  + Mapa de la ubicación.
  + Descripción de la empresa.
  + Servicios ofrecidos.

## Identificar los requerimientos funcionales

* Funcionalidades según el tipo de usuario presente.
* Área de acceso privado para administrador.
  + ABM para los productos publicados.
  + ABM para los usuarios registrados.
* Carrito de compras (fuera de alcance en esta oportunidad).
* Formularios de contacto.

# Definición de la estructura del sitio

## Mapa del sitio

|  |
| --- |
| Encom |
| API JAVA |
| Segunda Parte |

# Definicion de los limites del sistema

En esta sección se describirá a grandes rasgos las funcionalidades que cubrirá el sistema y las que quedan fuera del alcance del mismo. Para simplificar la documentación, se presentarán Diagramas de Secuencia de Sistema (DSS) en reemplazo de Casos de Uso.

## Requerimientos generales

Quedan dentro de los límites del sistema los siguientes requisitos u objetivos que se esperan que cumpla el sistema o el sitio en general:

* Identificación y Autenticación de clientes/usuarios.
  + Usuario Gest (invitado)
  + Cliente registrado
  + Administrador
* CRUD de clientes registrados.
* CRUD de productos.
* Envío de consultas por mail al web master.

Quedan fuera del alcance/límite de este proyecto los siguientes:

* Shopping Cart (Carrito de compras).
* Modelado del control de inventario y stock.
* Modelado de la facturación.
* Modelado de pago y medios de pago.
* Implementación de la seguridad en la transmisión de datos sensibles (al menos hasta el momento de redacción de este informe).

# Estructura del Sistema

## Diagrama de bloque

Se presenta la estructura sugerida a seguir y la definición de las capas de la aplicación.

**<VIEW>**

JSP + HTML

**<CONTROLLER>**

Servlet

(Principio de Rest API)

**<MODEL>**

Business Logic

**<DAO>**

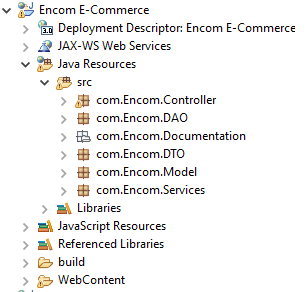
DB

Implementación del patrón MVC (Modelo-Vista-Controlador) y además de una capa DAO (Data Access Object) para obtener transparencia e independencia a la hora de acceder a los datos. Brindando robustez ya que da la posibilidad de implementar varias bases de datos y facilita la migración de las mismas.

Otro de los pilares sobre los que se basa este sistema es una aproximación a la arquitectura REST (Representational state transfer - Transferencia de Estado Representacional) que implementa una interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc) por medio de verbos (GET, POST, PUT, DELETE). Aunque no se use un formato de intercambio de información independiente, este sistema esta pensado para posteriormente utilizar JSON(JavaScript Object Notation) como medio de intercambio de datos para hacer independiente el front-end y el back-end. La idea es que ni el cliente ni el servidor necesiten recordar ningún estado de las comunicaciones entre mensajes.

# Estructura de directorios

La estructura de directorios del proyecto es la siguiente, donde se crearon paquetes con el prefijo com.Encom.\* para separar las capas, documentación y services.



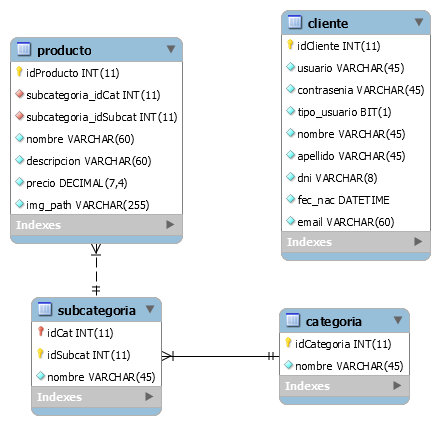
# Base de datos

La implementación de la base de datos se realiza en MySQL mediante Workbench.

## Diagrama entidad-Relación

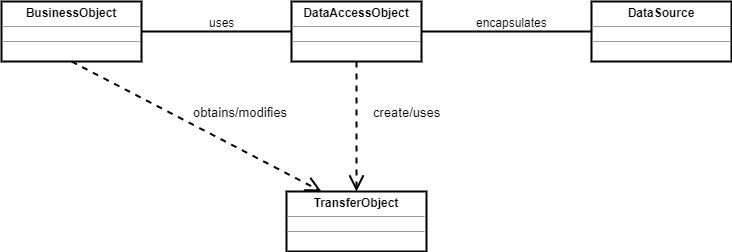
Diagrama EER de la base de datos, cuatro tablas principales. No existen triggers, store procedures ni functions.

Para las operaciones CRUD realizadas con el esquema, se creo un usuario de la DB con nombre de usuario “user”, el cual solo tiene los privilegios de create, update, delete, select y file.

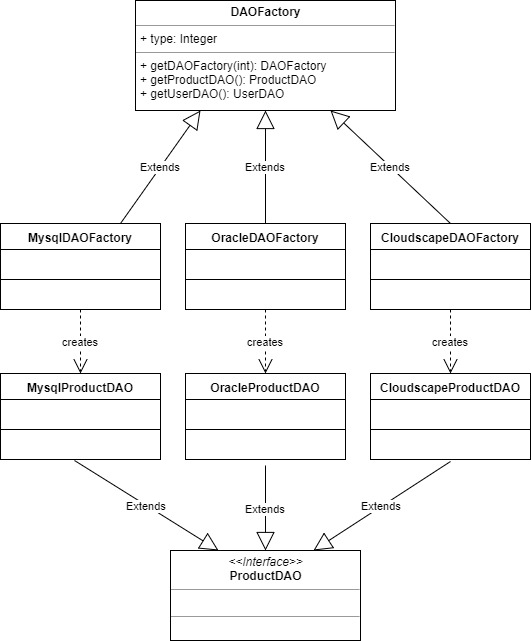


# Capa Dao

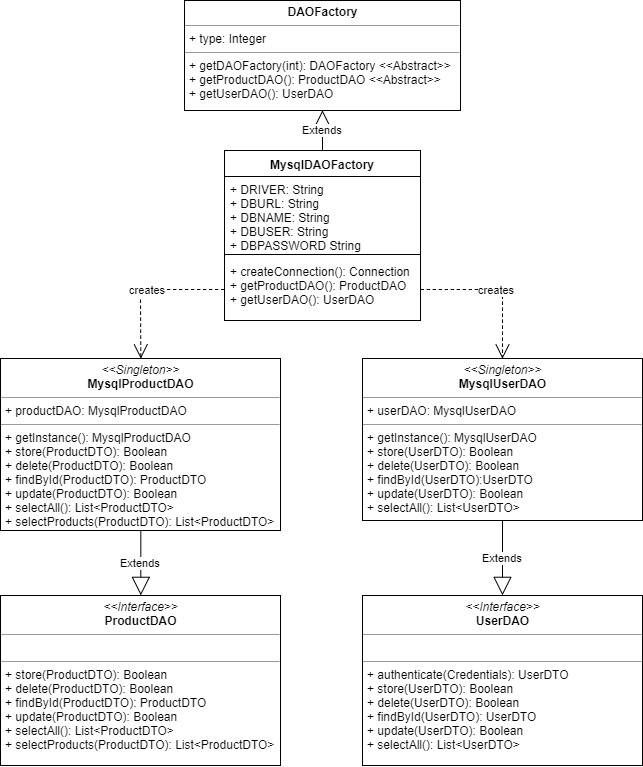
El concepto de Data Access Object (DAO) se basa en el siguiente diagrama de clases:



Aplicando el patrón de diseño Abstract Factory podemos lograr encapsular la lógica e implementación especifica de las operaciones y sintaxis SQL de cada motor de Gestión de Base de Datos. Como vemos en la siguiente figura a modo de ejemplo, podemos transferir y manipular los datos desde la capa del modelo de negocio hacia la capa DAO de forma transparente y robusta. Podemos discriminar que datos guardar en cada base de datos, logar una fácil migración de los mismos con solo crear las clases e interfaces correspondientes a esa implementación o mantener una redundancia en dos recursos distintos.



A continuación, se representa la estructura de clases y herencias actuales en el sistema, aunque podría implementarse el esquema anterior sin más problemas.



# Diagramas de Diseño

A continuación, se proveen los Diagramas de Secuencia de Sistema (DSS en reemplazo de los Caso de Uso de Usuario), Diagramas de Secuencia de Diseño (DSD) y Diagramas de Herencia Java para el CRUD de Productos.

## CRUD Productos

*Ver en Anexo I.*

# diagrama de herencia

*Ver en Anexo II.*

# código fuente

Se provee del código fuente del CRUD de productos, ya que se considera el core business de este sistema y es el objeto de estudio principal que representa la interacción entre todas las capas construidas. Además, también está incluido es código pertinente al manejo de archivos/imágenes (File Upload).

*Ver en Anexo III.*

# Annotations y xml

Para especificar las rutas (URL) de los Servlets (Controladores) se especificaron mediante Java Annotations ya que al tratarse de un sistema pequeño es más manejable y sencillo de usar.

Se utilizo la notación XMl para definir las error pages dictadas a continuación.

# http error pages

En adición, se especifican en el archivo web.xml las welcome file list y las error pages customizadas según las necesidades del sitio y las más comunes que puedan surgir, para así informar al cliente de forma más amigable las situaciones que puedan surgir mediante códigos HTTP.



# endpoints

Ver anexo IV.

# empaquetamiento y despliegue

El proyecto se exporto de eclipse mediante el formato WAR ( Web Application Archive - Archivo de aplicación web) es un archivo JAR utilizado para distribuir una colección de JavaServer Pages, servlets, clases Java, archivos XML, bibliotecas de tags y páginas web estáticas (HTML y archivos relacionados) que juntos constituyen una aplicación web.

Se desplego en el hosting Jelastic “Save in Cloud”, su tiempo de prueba es de catorce días a partir del 30/11/2017.

**Usuarios:**

* **Administradores:**
  + Usuario: admin
  + Contraseña: encom1234
* **Clientes:**
  + Usuario: john
  + Contraseña: doe

Ambos tipos de usuario tienen un time-out de 15 minutos.

**Dominio:** <http://encom.jelastic.saveincloud.net/>

# Conclusión

Como conclusión se debe hacer referencia en los pilares de mantenibilidad y robustez sobre los que se basó este desarrollo. Primeramente, por el modelado de una estructura modularizada de capas que brinda el modelo MVC, sumado de una cuarta capa que es la Capa DAO, brinda una alta mantenibilidad del código y del modelo de negocio, permitiendo una adaptación ante un cambio de forma versátil. La definición de la estructura de la base de datos y el modelado con patrones de diseño da la posibilidad de utilizar varias DB encapsulando las implementaciones. En segundo lugar, se prosiguió a realizar los diagramas de diseño de una parte del core business que es el manejo de productos. Luego la definición de los endpoints e interfaces realizadas con Boostrap, html5, css propio y el añadido de Javascript para el comportamiento en algunas páginas. Finalmente, la implementación en código JAVA de lo anteriormente descripto, el empaquetamiento en un archivo WAR y el despliegue en el servidor web. Podemos concluir que el sitio alcanzo de manera satisfactoria los objetivos planteados en un inicio.