MistiStore

Documento de Arquitectura de Software

Versión 0.4

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 11/06/12 | 0.1 | Apertura de la versión inicial del documento. | Franci Suni Lopez |
| 11/06/12 | 0.2 | Agregados diagrama de arquitectura de alto nivel. | Rommel Anatoli Quintanilla Cruz |
| 15/06/12 | 0.3 | Revisión del documento. | Prof. Robert Arisaca  Rommel Anatoli Quintanilla Cruz |
| 17/06/12 | 0.4 | Modificación de las vistas, y del diseño en capas. | Franci Suni Lopez |

Tabla de contenidos

1.- Introducción 4

1.1- Propósito 4

1.2- Alcance 4

1.3.- Definiciones, Acrónimos y Abreviaturas 4

2.- Objetivos y restricciones de la arquitectura 5

3.- Vista de Casos de Uso 6

3.1- Diagrama de Casos de Uso 6

4.- Vista Lógica 6

4.1- Diagrama de Clases de Diseño 6

5.- Vista de Implementación 7

5.1.- Capa de Aplicación Externa 7

5.2.- Capa de Aplicación Interna 8

5.2.1 Diseño en Capas 8

6.- Vista de Despliegue 9

7.- Calidad 10

Plan de Arquitectura de Software

# 1.- Introducción

## 1.1- Propósito

El presente documento provee una visión inicial para la arquitectura de la aplicación Tienda Virtual MistiStore. De esta manera, se busca capturar y asentar las decisiones importantes que serán tomadas en el desarrollo de la aplicación.

## 1.2- Alcance

Se muestra a alto nivel el diseño de la arquitectura por vistas de la aplicación. En cada una, se presentan los diagramas correspondientes, a saber: modelo conceptual, diagrama de clases, casos de uso, diagramas de interacción, entre otros.

## 1.3.- Definiciones, Acrónimos y Abreviaturas

ARQUITECTURA DE SOFTWARE: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

STAKEHOLDER: Individuo, equipo u organización con intereses relativos al sistema.

MODULO O COMPONENTE: cualquier elemento estructural abstracto, visible, externo, de alto nivel, analizable, que pueda constituir una funcionalidad de la solución del sistema.

INTERNET: Infraestructura de redes a escala mundial que se conecta a la vez a todo tipo de computadores. Desarrollado originariamente para los militares de Estados Unidos, después se utilizó para el gobierno, la investigación académica y comercial y para comunicaciones.

COMERCIO ELECTRONICO: Entrega de información, productos, servicios o pagos por medio de líneas telefónicas, redes de ordenadores o de cualquier otro dispositivo electrónico.

JAVA SERVER PAGES (JSP) es una tecnología [Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) que permite generar contenido dinámico para web, en forma de documentos [HTML](http://es.wikipedia.org/wiki/HTML), [XML](http://es.wikipedia.org/wiki/XML) o de otro tipo.

MODELO VISTA CONTROLADOR (MVC) es un patrón de [arquitectura de software](http://es.wikipedia.org/wiki/Arquitectura_de_software) que separa los [datos](http://es.wikipedia.org/wiki/Datos) de una aplicación, la [interfaz de usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario), y la [lógica de negocio](http://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) en tres [componentes](http://es.wikipedia.org/wiki/Componentes) distintos.

FRAMEWORK: Representa una [arquitectura de *software*](http://es.wikipedia.org/wiki/Arquitectura_de_software) que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

# 2.- Objetivos y restricciones de la arquitectura

El desarrollo de la aplicación se enfoca en que llegue a tener características que sean sostenibles, eficientes y fáciles de usar para cualquier usuario del sistema.

Nuestras principales metas a nivel de arquitectura son las siguientes:

• Performance: El desempeño de la aplicación debe ser muy eficiente de tal manera el usuario inmediato y todos los demás observen lo más rápidamente posible los cambios realizados en un momento determinado.

• Usabilidad: El diseño debe ser orientado por y para la comodidad del usuario, de manera que la interfaz sea intuitiva y fácil de manejar, al mismo tiempo que se fomente altamente la interacción entre ambos. De la misma forma, el usuario debe tener la capacidad de equivocarse y regresar a un estado seguro en el que se le permita cumplir con su objetivo original sin que se le haga tedioso o complicado el proceso para llegar a dicho fin.

• Interactividad: la tecnología funciona a través de la interacción con el usuario. Los consumidores entablan un diálogo que ajusta en forma dinámica la experiencia para el individuo, y hace del consumidor un coparticipante en el proceso de entrega de bienes en el mercado.

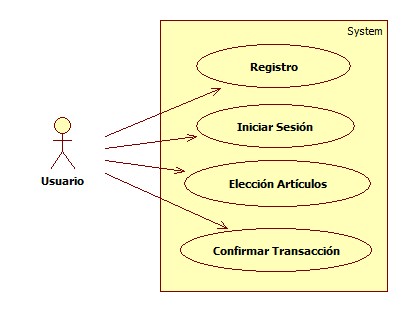
• Tecnología social: generación de contenido por parte del usuario en redes sociales. Los nuevos modelos sociales y los negocios de internet permiten que el usuario cree y distribuya su propio contenido, y que estas sean soportadas por las redes sociales.

En la planificación de este proceso hemos encontrado las siguientes restricciones:

1. Restricciones de contenido: debido a que nuestro sistema está basado y propuesto inicialmente para el mercado peruano, el mismo debe estar de acuerdo al marco legal vigente comercial.
2. Restricciones de tecnología y uso de herramientas de desarrollo: están predefinidos los instrumentos a utilizar así como también la plataforma tecnológica sobre la que se va a desarrollar el sistema. La aplicación será implementada usando JSP, es por ello que las herramientas utilizadas estarán determinadas por las funcionalidades ofrecidas por dicho lenguaje.

# 3.- Vista de Casos de Uso

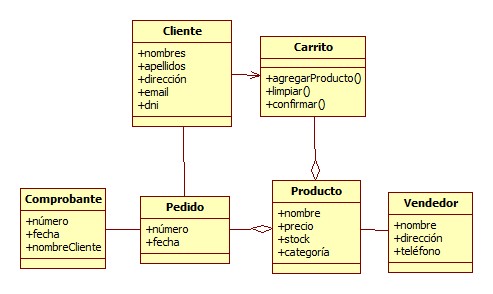
## 3.1- Diagrama de Casos de Uso



# 4.- Vista Lógica

## 4.1- Diagrama de Clases de Diseño

Se tiene el siguiente Modelo Conceptual donde se dan las asociaciones pertinentes entre los conceptos más relevantes para la aplicación Tienda Virtual MistiStore.



# 5.- Vista de Implementación

Usar Google App Engine como plataforma trae consigo beneficios como usar la misma infraestructura que usa Google para sus productos, es decir la aplicación soportará grandes volúmenes de datos y concurrencia, dispondrá de esquemas de resguardo de información, y alta disponibilidad asegurada.

## 5.1.- Capa de Aplicación Externa

A partir de la arquitectura presentada, que está basada fundamentalmente en el patrón MVC, se diseñaron los paquetes que interrelacionados implementan la funcionalidad del sistema.

Las categorías existentes son las tres definidas en el modelo MVC (modelo, vista y controlador) más una cuarta denominada infraestructura. Esta cuarta categoría es provista por el entorno App Engine e implementa los servicios de base que permitan desarrollar una aplicación dentro de la arquitectura. Algunos de los servicios dentro de la categoría son:

• Persistencia de entidades.

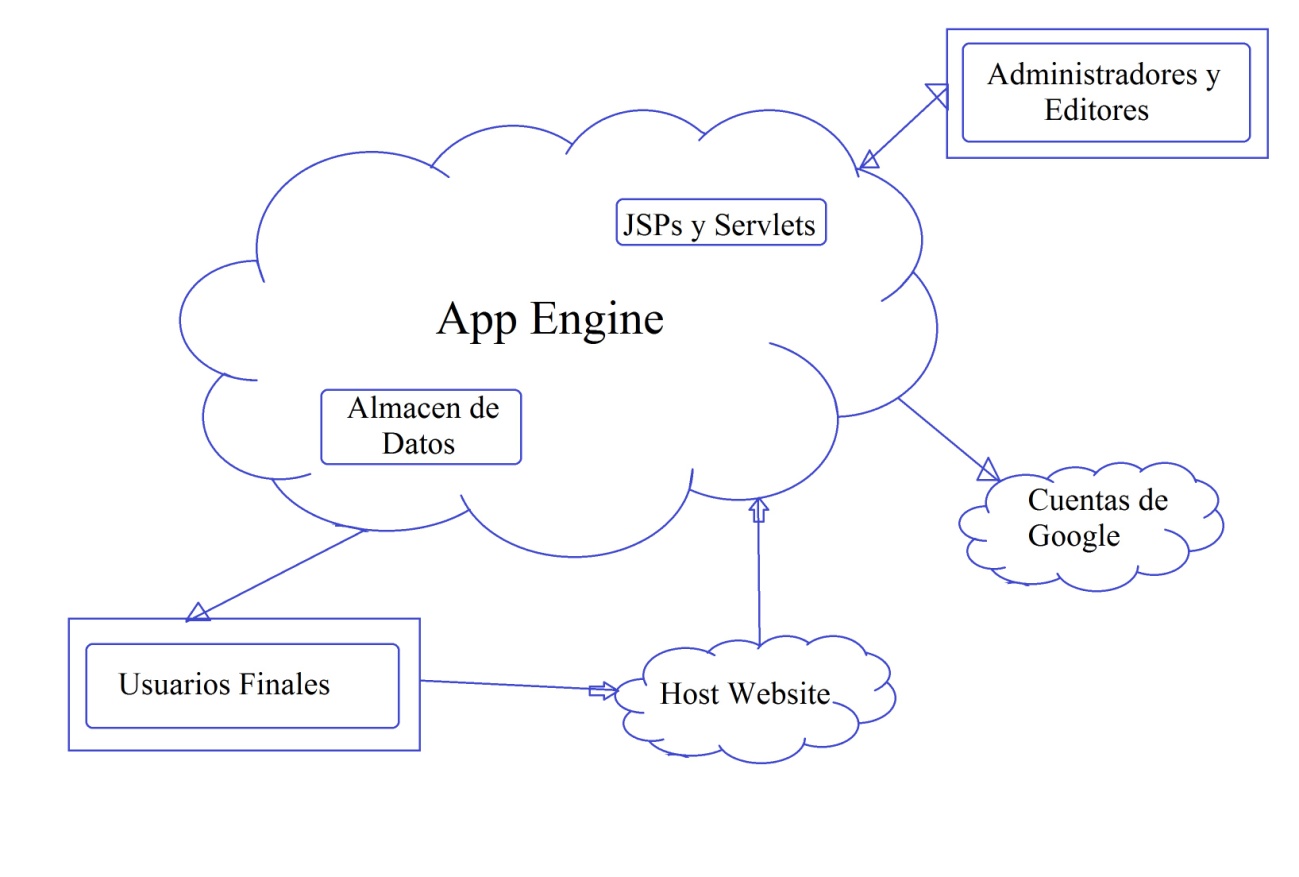
• Acceso a archivos del sistema operativo.

• Motor de plantillas para representación de página web dinámicas.

• Autenticación de usuarios

Estos servicios son parte de la infraestructura de la aplicación y, generalmente son consumidos por una o más de las tres categorías de MVC.

A continuación mostramos un marco general del funcionamiento del sistema.

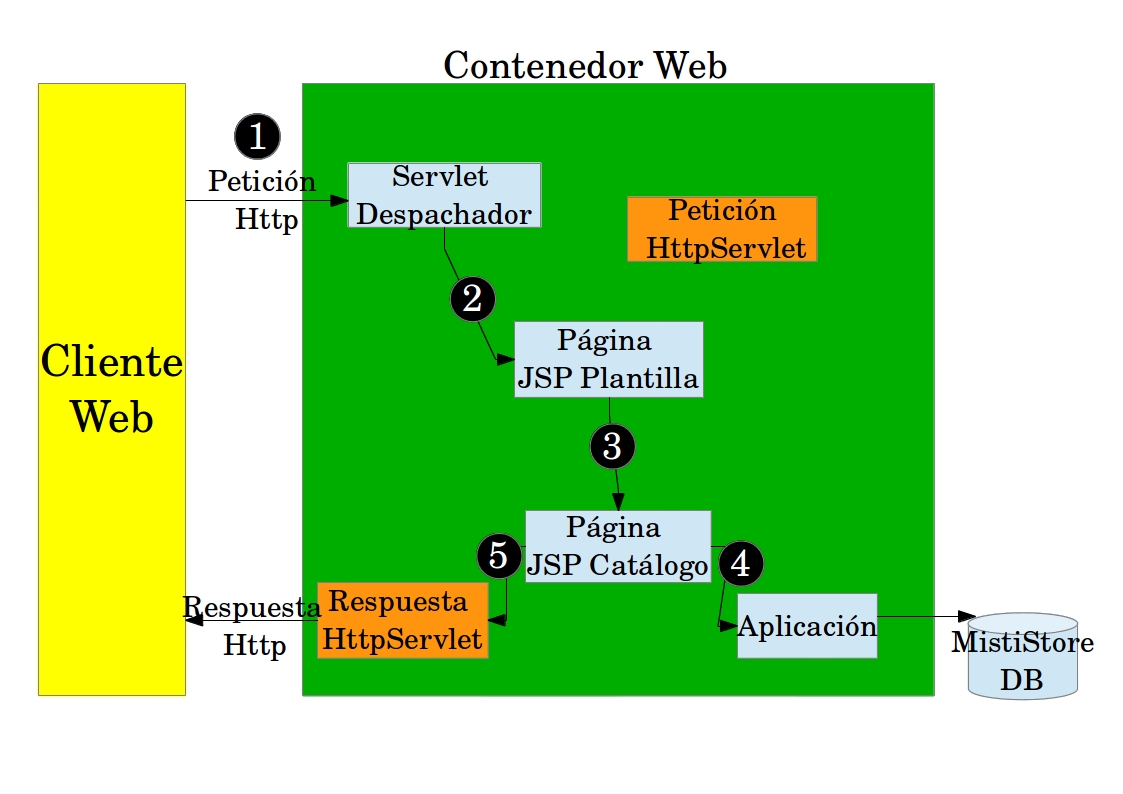


**Figura 1.** Vista lógica general(Fuente: Elaboración propia)

## 5.2.- Capa de Aplicación Interna

### 5.2.1 Diseño en Capas

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema.



**Figura 2.** Diseño en Capas(Fuente: Elaboración propia)

a) Capa Interfaz de Usuario

Esta capa contiene el paquete Interfaz de Usuario, el cual almacena todas las clases con las cuales el usuario puede interactuar como lo son las ventanas.

b) Capa Lógica del Negocio

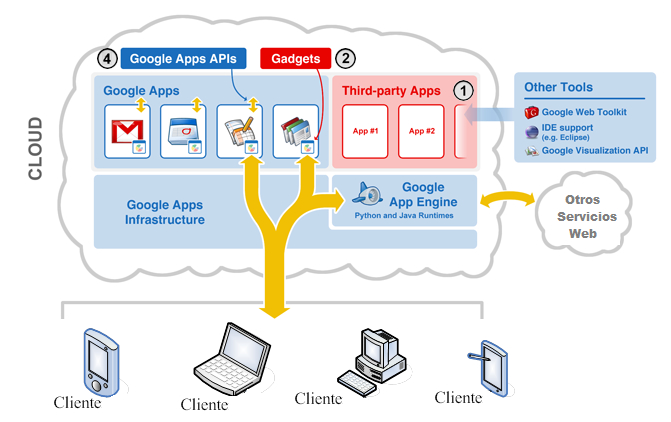
Esta capa contiene los paquetes de Servicios de Negocio y Entidades de Negocio. Contiene la lógica para el manejo de las operaciones del negocio.

c) Capa Persistencia

Esta capa contiene el paquete de Objetos de Acceso de Datos, que brinda una interfaz transparente para la interacción con el Framework (Google App Engine) el cual enviará al Driver el conjunto de sentencias para interactuar con la Base de Datos.

# 6.- Vista de Despliegue

La siguiente figura muestra la vista de despliegue, basada en la infraestructura que Google App Engine nos proporciona.



**Figura 3.** Vista de Despliegue(Fuente: <http://www.smart-traveler.us/project-vision>)

# 7.- Calidad

La arquitectura de software apoya las exigencias de calidad.

1. La interfaz de usuario será WEB.
2. La interfaz de usuario del Sistema MistiStore será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal.
3. Cada despliegue de opciones de pantalla, tendrá la ayuda en línea para el usuario. La ayuda En línea incluirá paso a paso instrucciones en la utilización del Sistema. La ayuda En línea incluirá definiciones para términos y acrónimos.