Ing. Eric Romero Martínez

4 de abril de 2016

Posgrado en Ciencias e Ingeniería de la Computación

Tecnología Orientada a Objetos.

Documento de diseño

Versión 1.0



Resumen.

En el presente documento se indica la arquitectura de software para el desarrollo del proyecto “Presta un libro”, incluyendo la descripción de los paquetes, componentes, despliegue y clases.

Tabla de contenido

[1. Representación de la Arquitectura 1](#_Toc448089156)

[1.1 Definición del ambiente de implementación 1](#_Toc448089157)

[2. Vista Lógica 1](#_Toc448089158)

[2.1 Descripción 1](#_Toc448089159)

[2.2 Paquetes de la arquitectura 1](#_Toc448089160)

[2.2.1 Paquete: Modelo 2](#_Toc448089161)

[2.2.2 Paquete: Vista 3](#_Toc448089162)

[2.2.3 Paquete: Controlador 3](#_Toc448089163)

[2.3 Identificación de las clases 3](#_Toc448089164)

[2.3.1 Clases de Interfaz 4](#_Toc448089165)

[2.3.2 Clases de entidad 4](#_Toc448089166)

[2.3.3 Clases de control 4](#_Toc448089167)

[3. Vista de Despliegue 4](#_Toc448089168)

[3.1 Descripción 4](#_Toc448089169)

[3.2 Modelo de despliegue 4](#_Toc448089170)

[4. Vista de Datos 4](#_Toc448089171)

[4.1 Descripción 4](#_Toc448089172)

[4.1.1 <Diagrama de la Base de Datos> 4](#_Toc448089173)

[5. Vista Dinámica 5](#_Toc448089174)

[5.1 Descripción 5](#_Toc448089175)

[5.2 Diagramas de secuencia 5](#_Toc448089176)

[5.3 Diagramas de navegación 5](#_Toc448089177)

Documento de Diseño de Software

# Representación de la Arquitectura

El patrón de arquitectura a utilizar es el Modelo Vista Controlador, el cual es un patrón que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Los atributos de calidad del software son:

Funcionales

* El despliegue de cada página debe ser menor a 3 segundos.
* El uso del sistema debe ser intuitivo.
* Los requerimientos funcionales y no funcionales están separados.
* La información personal de cada usuario permanece oculta ante otros usuarios.
* Las vistas contienen un contraste lo suficientemente bueno que facilita la lectura.

No funcionales

* El sistema debe estar elaborado con la arquitectura Modelo Vista Controlador
* El software deberá ser desarrollado con Apache Tomcat.

Se utilizará la arquitectura Modelo Vista Controlador ya que es parte de los requerimientos no funcionales solicitados por el cliente.

## Definición del ambiente de implementación

|  |  |  |
| --- | --- | --- |
| **Concepto** | **Herramienta** | **Versión** |
| Lenguaje de programación | Java (Tomcat) | 7 |
| Framework | Hibernate. | 5.0.9 |
| Diagramador UML | Argo UML. | 0.34 |
| Manejador de bases de datos | Mysql. | 5.6 |
| Servidor de aplicaciones | Mozilla Firefox. | 45 |
| IDE | Eclipse | 4.5.2 |
| Control de versiones | Git | 2.7.2 |

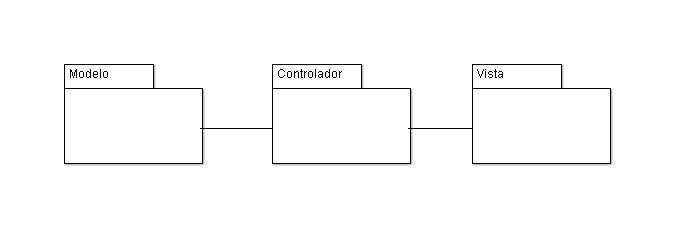
# Vista Lógica

## Descripción

 El modelo está compuesto por los paquetes Modelo, Vista y Controlador. En la siguiente sección se describe el comportamiento, características y detalles de cada uno de ellos.

## Paquetes de la arquitectura

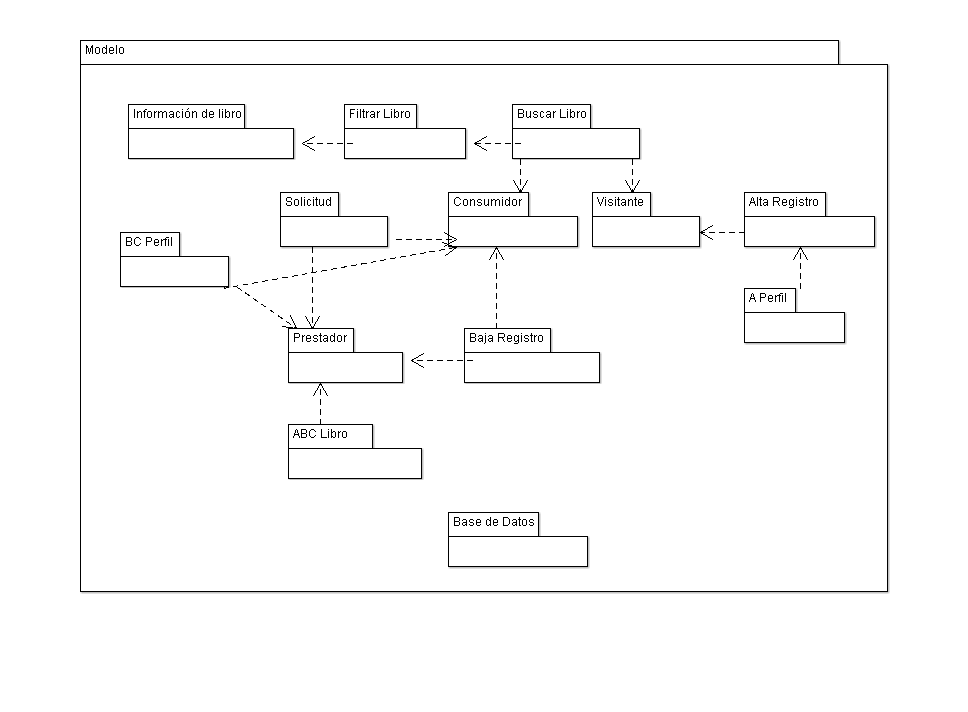
En la siguiente figura se muestra la organización de los paquetes Modelo, Vista y Controlador.



### Paquete: Modelo

#### Descripción

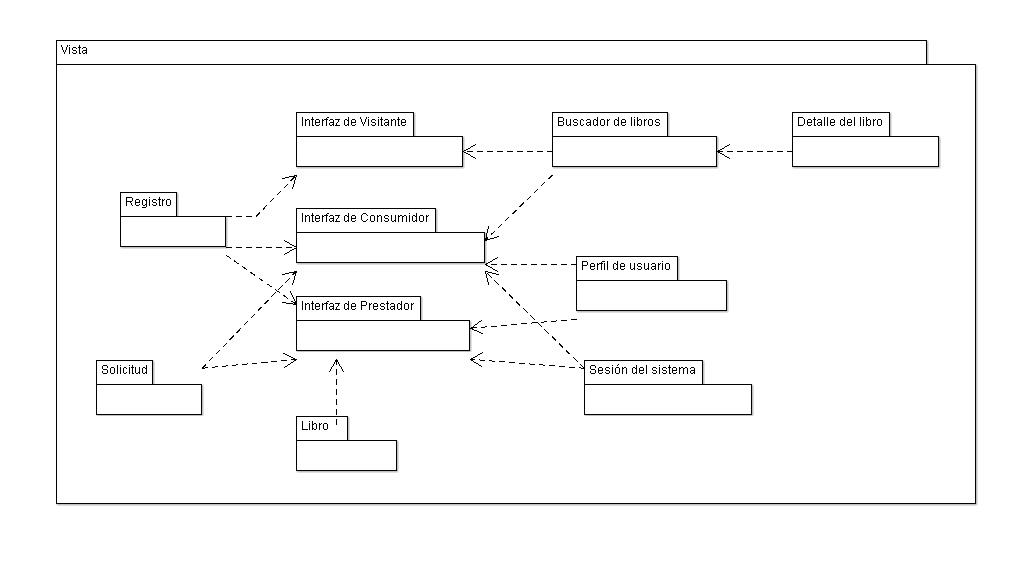
Mantiene la comunicación con la base de datos, con lo que puede recuperar y almacenar los datos, así como procesarlos y adecuarlos para ser utilizados por otro paquete.



### Paquete: Vista

#### Descripción

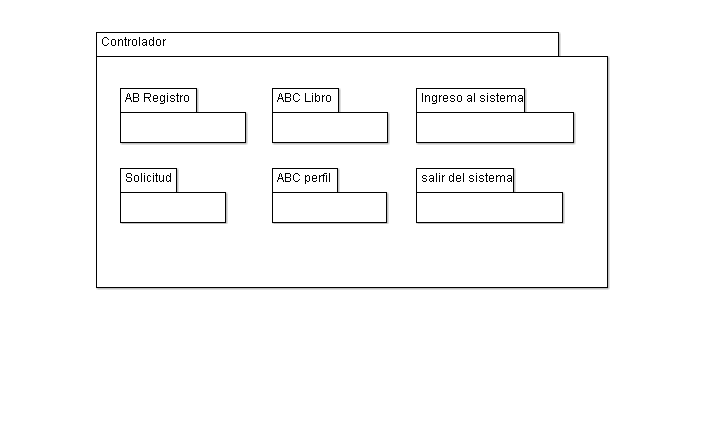
Esta encargada de desplegar la información solicitada al usuario.



### Paquete: Controlador

#### Descripción

Se encarga de recibir las solicitudes del usuario y reenviar las mismas hacia las vistas para que se le presente al usuario la información deseada, en algunas ocasiones el controlador solicita al modelo que le recupere o almacene información de la base de datos para después reenviarle la información a la vista.



## Identificación de las clases

 [Esta sección contendrá los diagramas que describen el comportamiento estático del sistema y los tipos de relaciones existentes entre las clases. Se generará un diagrama de clases por cada paquete que contenga el sistema.

Para generar el documento, es conveniente como primer paso identificar qué clases será necesario crear para que el sistema cumpla con los requerimientos establecidos, esto se realizará revisando cada caso de uso identificando qué necesidades cubre cada uno de éstos y con ello decidir qué clases cumplirán con ese trabajo.]

### Clases de Interfaz

[.]

### Clases de entidad

[.]

### Clases de control

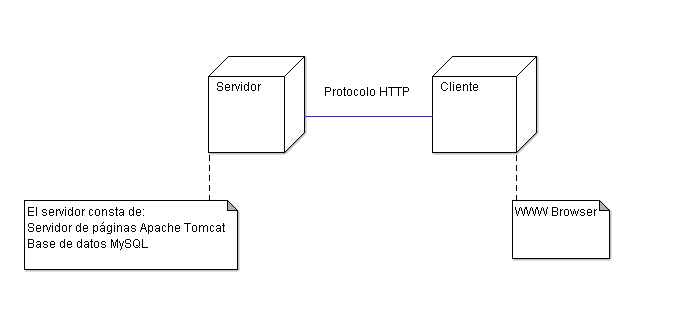
[.]

# Vista de Despliegue

## Descripción

El software funciona en ámbito local, por lo que se requiere de un solo nodo servidor que contendrá Apache Tomcat para las aplicaciones web y el Sistema Administrador de Base de Datos MySQL.

## Modelo de despliegue



# Vista de Datos

## Descripción

 [Debe incluirse una descripción del mecanismo de almacenamiento de datos del sistema. Esta sección es opcional, si hay poca o ninguna persistente de datos, o la traducción entre el modelo de diseño y el modelo de datos es trivial.]

### <Diagrama de la Base de Datos>

# Vista Dinámica

## Descripción

[Cada caso de uso deberá ser representado con un diagrama de secuencia, éstos muestran cómo se comportan los objetos entre ellos a través del tiempo. Representarán el comportamiento dinámico de los casos de uso Deberá tomarse en cuenta que por cada caso de uso deben generarse diagramas de secuencia ejemplificando los flujos normales y aquellos que manejan excepciones, flujos excepcionales, además se deberá identificar cada una de las clases participantes en el caso de uso para que se representen en el diagrama de secuencia.

El diagrama de estados, indicará cómo navega el usuario a través de las interfaces del sistema. Cada diagrama tendrá un estado inicial que se identificará con una circunferencia negra que rodea un punto negro, los estados serán representados por un óvalo con el nombre de la interfaz en la que se encuentra el usuario al realizar alguna acción, las acciones estarán determinadas en el documento por flechas con un estado origen, una acción y el estado destino. Es necesario indicar aquellos eventos que generarían un error y manejarlos de acuerdo a sus necesidades, por ejemplo enviando al usuario a un estado donde pueda recuperarse para continuar sin necesidad de reiniciar todo el proceso.]

## Diagramas de secuencia

[Diagramas de secuencia.]

## Diagramas de navegación

[Diagramas de estado.]