Página Web construcciones Monsa

Version <2.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 14/09/2015 | 1.0 | Esta es la primera interacción con la plantilla arquitectura, donde se va a plasmar las ideas iniciales. | Deiby Fabián Loaiza Mora  Felipe Andres Jamioy Girón |
| 28/09/2015 | 2.0 | En esta versión del documento se harán las correcciones de la primera versión, consolidando la arquitectura.   * Casos de uso * Diagrama de despliegue * Métodos de calidad. | Deiby Fabián Loaiza Mora  Felipe Andres Jamioy Girón |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 4

5. Logical View 4

5.1 Overview 4

5.2 Architecturally Significant Design Packages 4

5.3 Use-Case Realizations 4

6. Process View 4

7. Deployment View 4

8. Implementation View 4

8.1 Overview 4

8.2 Layers 4

9. Data View (optional) 4

10. Size and Performance 4

11. Quality 4

# Introduction

*En esta plantilla se tendrá un análisis de la arquitectura de software, donde se presentara una propuesta elaborada de la arquitectura que tendrá la página web “Construcciones Monsa”, como la implementación que se tendrá con los requisitos funcionales y no funcionales.*

## Purpose

*En este documento se definirá la arquitectura que tendrá la página web “Construcciones Monsa”, donde se utilizara el modelo “4+1 vistas” diseñado por el profesor* [*Philippe Kruchten*](http://philippe.kruchten.com/)*, donde por medio de estas vistas se tendrá un mayor acercamiento a la arquitectura que se necesita.*

*Se utilizara diagramas de despliegue, diagrama de componentes, diagrama de caso de uso y un diagrama de clases para así poder tener un acercamiento de lo que se quiere a nivel de software y poder comprender los requisitos de este.*

## Scope

Se utilizara el modelo “4+1 vistas” para hacer una abstracción de los artefactos necesitados para realizar la arquitectura de software de “Construcciones Monsa”, utilizando diagrama con UML los cuales son:

* *Diagrama de casos de uso*
* *Diagrama de clases*
* *Diagrama de despliegue*
* *Diagrama de componentes*

*Con estos diagrama podemos tener una visión más clara de la arquitectura de software que se quiere y como se va a relacionar ciertos estándares de calidad con el aplicativo.*

## Definitions, Acronyms, and Abbreviations

**UML:** Lenguaje Unificado de Modelado es el lenguaje de [modelado](https://es.wikipedia.org/wiki/Modelado) de sistemas de [software](https://es.wikipedia.org/wiki/Software) más conocido y utilizado en la actualidad; está respaldado por el [OMG](https://es.wikipedia.org/wiki/Object_Management_Group) (Object Management Group).

***Vista de despliegue:*** *En esta vista se muestran los procesos que hay en el sistema y la forma en la que se comunican estos procesos; es decir, se representa desde la perspectiva de un*integrador de sistemas*, el flujo de trabajo paso a paso de negocio y operacionales de los componentes que conforman el sistema. Para completar la documentación de esta vista se puede incluir el diagrama de actividad de UML.*

***Vista de escenario:*** *Esta vista va a ser representada por los casos de uso  software y va a tener la función de unir y relacionar las otras 4 vistas, esto quiere decir que desde un caso de uso podemos ver cómo se van ligando las otras 4 vistas, con lo que tendremos una trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso. Para completar la documentación de esta vista se pueden incluir los diagramas de casos de uso de UML.*

***Vista de implementación:*** *En esta vista se muestra desde la perspectiva de*un ingeniero de sistemas*todos los componentes físicos del sistema así como las conexiones físicas entre esos componentes que conforman la solución (incluyendo los servicios). Para completar la documentación de esta vista se puede incluir el diagrama de despliegue de UML.*

***Vista de despliegue:*** *En esta vista se muestra el sistema desde la perspectiva de*un programador*y se ocupa de la gestión del software; o en otras palabras, se va a mostrar como es*tá *dividido el sistema software en componentes y las dependencias que hay entre esos componentes. Para completar la documentación de esta vista se pueden incluir los diagramas de componentes y de paquetes de UML.*

***Vista lógica:*** *En esta vista se representa la funcionalidad que el sistema proporcionara a los*usuarios finales.*Es decir, se ha de representar lo que el sistema debe hacer, y las funciones y servicios que ofrece. Para completar la documentación de esta vista se pueden incluir los diagramas de clases, de comunicación o de secuencia de UML.*

***SQL Server:*** *Base de datos.*

## References

### Modelo “4+1” vistas de Kruchten (para Dummies)

Cocomo-ii.software.informer.com

Alex sobrino blog personal.

<http://blog.0balla.net/2012/05/escalabilidad-y-alta-disponibilidad-en-servidores-web/>

Alberto Fernández capel.

<http://codigocomestible.com/2010/03/28/mitos-escalabilidad-aplicaciones-web/>

## Overview

*En este documento se muestra las vistas del sistema en base a la arquitectura que se desea, por ello se dará evidencia de las diferentes vistas e incluso los patrones que afectaran y ayudaran la arquitectura.*

# Architectural Representation

*En este documento se va a mostrar detalladamente la arquitectura utilizando el modelo de “4+1 vistas” el cual es un modelo utilizado para describir la arquitectura del aplicativo Construcciones Monsa basado en múltiples vistas.*

***Vista lógica.***

*En esta vista lógica se va a representar la funcionalidad basada en el dominio del problema, utilizando la información suministrada por la empresa. Para esto se va a utilizar el diagrama de clases el cual nos va a facilitar una mejor compresión del negocio.*

*Diagrama: clases.*

***Vista de escenario.***

*En esta vista de escenario vamos a tener la función de unir las otras vistas por medio de un diagrama de caso de uso que nos va a ayudar a entender la trazabilidad que va a tener el aplicativo “Construcciones Monsa”.*

*Diagrama: casos de uso.*

***Vista de despliegue.***

*En esta vista de despliegue se va a mostrar el sistema desde la perspectiva de un programador es decir, del desarrollo a producción reflejando todos los componentes que se van a tener en cuenta para el aplicativo “Construcciones Monsa”.*

*Diagrama: componentes.*

***Vista de implementación.***

*En esta vista de implementación se va a basar los componentes físicos que se van a tener para el aplicativo “Construcciones Monsa”, donde se va a montar el proyecto y evidenciar su funcionalidad.*

*Diagrama: despliegue.*

# Architectural Goals and Constraints

*El objetivo de la arquitectura para el aplicativo “Construcciones Monsa” es asegurar la disponibilidad y la escalabilidad que va a tener en un futuro el aplicativo.*

* *Disponibilidad: Es importante para la empresa “Construcciones Monsa” tener alta disponibilidad ya que por esta plataforma va a fidelizar a sus futuros clientes, por lo tanto la disponibilidad de esta plataforma será fundamental ya que en el pasado han tenido problemas con sus plataformas.*
* *Rendimiento: Para “Construcciones Monsa” es importante que la plataforma web tenga un buen desempeño a la hora de poner en marcha este módulo que se va a implementar, ya que de no ser así los posibles compradores no se van a sentir a gusto y por lo tanto la empresa no lograra el objetivo de fidelizar a sus posibles clientes.*

***Requisitos***

* *Se debe visualizar los modelos de casa que se tiene para que el cliente pueda elegir uno a su gusto: La plataforma deberá mostrar al cliente las casas que la empresa diseña y sus atributos.*
* *La plataforma deberá calcular un valor inicial de la casa: Una vez el cliente haya elegido la casa de su gusto la plataforma calculara un valor inicial dependiendo el valor del dólar.*
* *Se debe hacer registro del cliente que quiera una cotización final: Una vez el cliente quiera hacer una cotización final deberá hacer un registro donde se tomaran sus datos personales y la forma de contacto de su preferencia.*
* *Se debe enviar al correo electrónico del cliente la cotización inicial con los componentes de la casa cotizada: Una vez el cliente se registre se le enviara un correo con los datos de la empresa, cotización inicial y las aclaraciones de cómo se contactar para una cotización final.*
* *Se debe hacer un registro de los clientes que hicieron una cotización final: Los clientes que se registren quedaran como personas que tomaron una cotización más detallada.*
* *Se debe tener un registro de los clientes que realizaron la compra de una casa: Una vez el cliente haga la compra y se firme el contrato el administrador de la página deberá cambiarle el estado al cliente a comprador.*

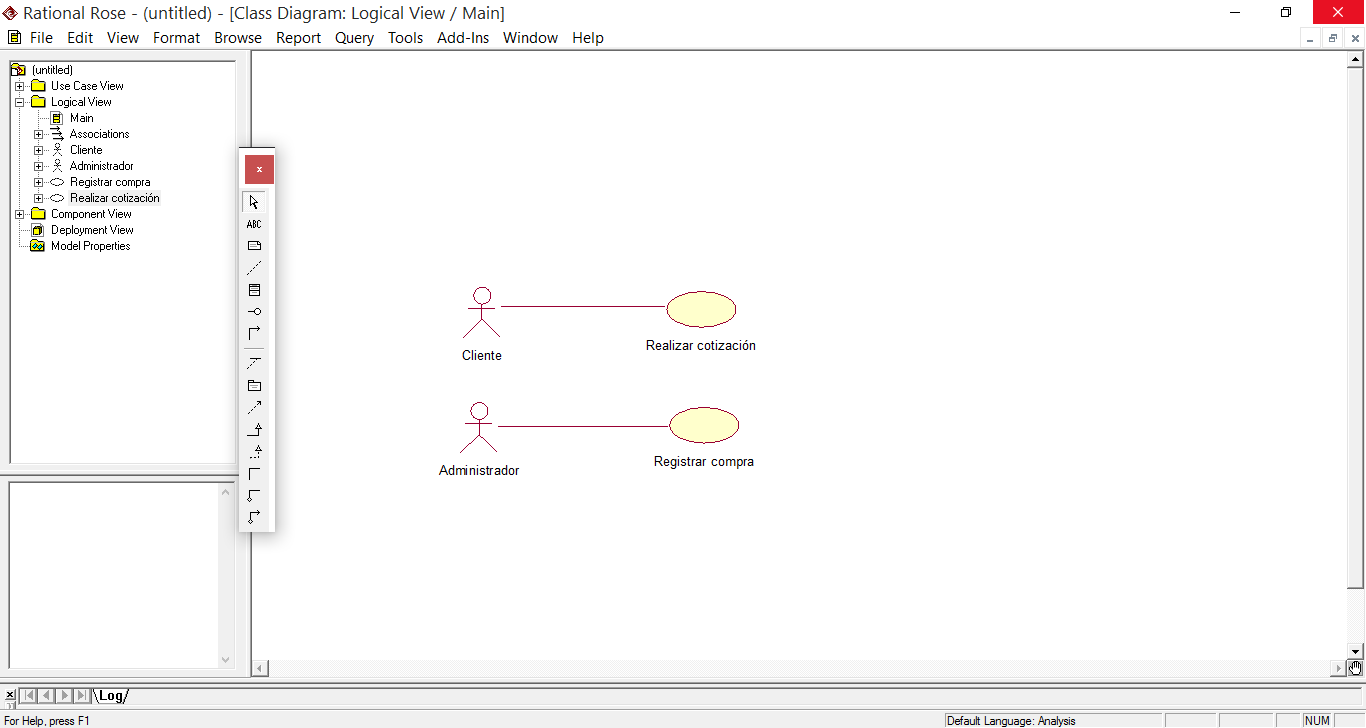
***Restricciones.***

*Hardware: Como la empresa es relativamente nueva las restricciones que tenemos es que a la hora de realizar las cotizaciones y la administración de la solo se tendrá un administrador y un equipo para este tipo de tareas.*

*Negocio: En un inicio se pensó que el cliente pudiera elegir las puertas y las ventanas de los modelos de casa que se tenían, pero la empresa no puede hacer este tipo de cosas ya que se dañaría el diseño original y por lo tanto el modelo.*

# Use-Case View

*En este diagrama se evidencia los casos de uso en los cuales podemos observar el comportamiento de estos.*



***Imagen 1.*** *Diagrama de caso de uso*

***Caso de uso.***

* ***Realizar cotización:*** *En este caso de uso se desea hacer una cotización inicial de un modelo de casa donde se tienen unas actividades específicas:*

1. ***Seleccionar casa:*** *Cuando el cliente se dirige a realizar una cotización lo primero que debe hacer es seleccionar el modelo de la casa que quiere para una posterior compra.*
2. ***Seleccionar tipo de casa:*** *Una vez el cliente elija el modelo deberá elegir las casas que hay con ese modelo.*
3. ***Calcular cotización:*** *Cuando el cliente seleccione la casa el sistema calculara el valor inicial estimado para esta propiedad.*
4. ***Registrar cliente:*** *Una vez el sistema calcula el valor, el cliente decide si seguir con el proceso y para eso deberá suministrar sus datos personales dando la forma de contacto que más le sea cómoda.*
5. ***Enviar cotización:*** *El sistema deberá enviar un correo electrónico al cliente con toda la información de la casa y los valores de la cotización.*

* ***Registrar compra:*** *En este caso de uso se va a registrar la compra de un cliente en cuando la compra se concrete, además también se podrá consultar los datos de los clientes.*

1. ***Consultar cliente:*** *Muestra la información del cliente y su estado actual (cotizante o comprador):*
2. ***Registrar cliente:*** *Una vez se haga la cotización y se concrete la compra de una casa, el administrador de la página deberá buscar al cliente y cambiar el estado de cotizante a comprador, esto con el fin de llevar unos datos que ayuden a crecer la empresa.*

# Logical View

# 

**Imagen 2.** Diagrama de clases

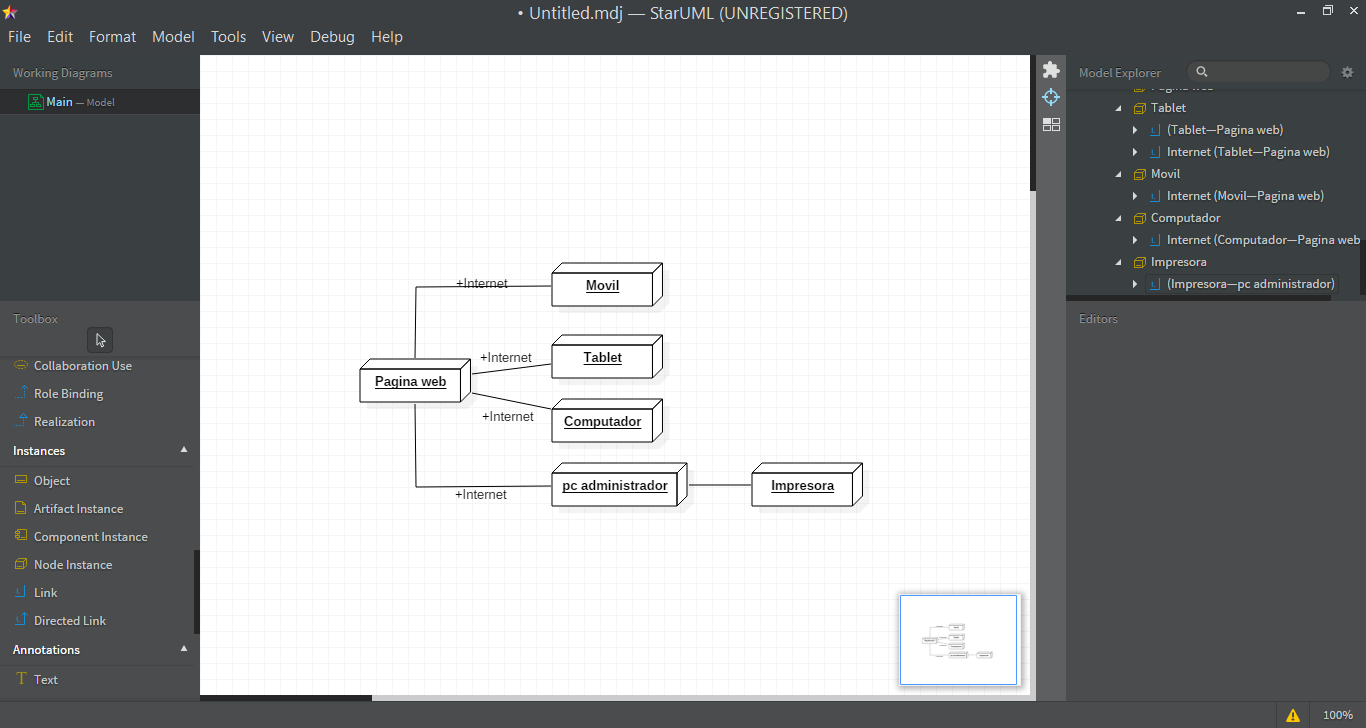
En este diagrama de clases podemos evidenciar el comportamiento de la empresa “Construcciones Monsa”, donde se puede evidenciar las clases que posteriormente se harán para el diseño del aplicativo, aunque es un diseño preliminar que se puede someter a cambios.

*La empresa va a constar de un administrador que se encargara se interactuar con los cambios de la página, también se va a manejar los datos de los usuarios que posteriormente se conviertan en clientes. Además los diseños o modelos de casa que se tengan tendrán una cotización inicial según el precio del dólar.*

# Process View

* *Disponibilidad: Es importante para la empresa “Construcciones Monsa” tener alta disponibilidad ya que por esta plataforma va a fidelizar a sus futuros clientes, por lo tanto la disponibilidad de esta plataforma será fundamental ya que en el pasado han tenido problemas con sus plataformas.*
* *Rendimiento: Para “Construcciones Monsa” es importante que la plataforma web tenga un buen desempeño a la hora de poner en marcha este módulo que se va a implementar, ya que de no ser así los posibles compradores no se van a sentir a gusto y por lo tanto la empresa no lograra el objetivo de fidelizar a sus posibles clientes.*

# Deployment View

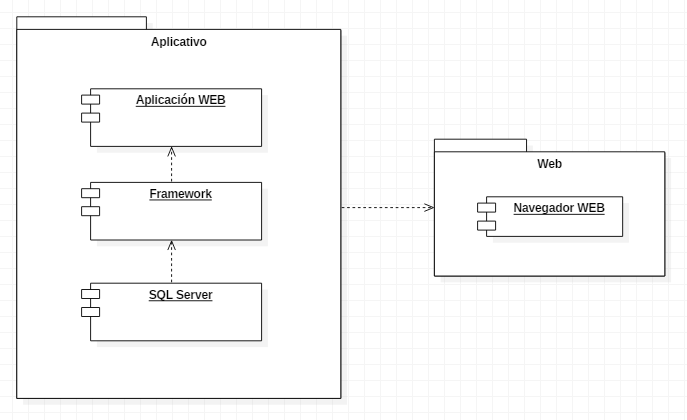


***Imagen 3.*** *Diagrama de despliegue*

*En la imagen 3 se puede observar los dispositivos en donde se puede desplegar el aplicativo “Pagina web Construcciones Monsa”.*

* ***Móvil:*** *Se puede desplegar en dispositivos móviles.*
* ***Tablet:*** *Se puede desplegar en tabletas.*
* ***Computador:*** *En los computadores no hay problema en el despliegue de la aplicación.*
* ***Pc administrador:*** *Se desplegara en el pc donde se va a administrar la página.*
* ***Impresora:*** *La impresora nos ayudara a imprimir los documentos necesarios para una cotización.*

# Implementation View



***Imagen 4.*** *Diagrama de componentes*

## Overview

## Layers

* *SQL Server: Es donde se le va a dar manejo a los datos de los clientes que realice una cotización y también a los que van a realizar una compra.*
* *Framework: Se utilizara un framework con el cual le vamos a dar manejo a la parte de la programación y esta conectara la base de datos con el aplicativo.*
* *Navegador web: Es el que permite el acceso a la página web, interpretando la información de los distintos tipos de archivos que tendremos en el aplicativo.*

# Size and Performance

*Para el proyecto CONSTRUCCIONES MONSA utilizaremos una serie de patrones que nos ayudarán a agilizar el desarrollo de la aplicación.*

## *Patrón de diseño Proxy:*

*Este patrón es ideal para nuestra aplicación web ya que la aplicación tendrá un alto contenido de imágenes con alta resolución que le permitan al cliente poder observar con detalle cada parte del diseño preliminar de su casa. Al contener tantas imágenes la aplicación web tardaría en cargarlas todas, es ahí donde el patrón de diseño Proxy se vuelve importante porque solo cargará la imagen con el diseño que el usuario final desee.*

## *El patrón de diseño Unit of Work (unidad de trabajo)*

*Se va a utilizar en el aplicativo web CONSTRUCCIONES MONSA con el fin de darle manejo a los datos que ingrese el cliente. Ya que puede ser que la base de datos falle y no se guarden los datos, esto garantiza que haya concurrencia a la hora de agregar, eliminar y actualizar los datos de futuros compradores.*

## *Patrón Lazy Loading (Carga Lenta)*

*Utilizaremos este diseño en la aplicación web porque nos permite cargar los objetos de la aplicación solo en el momento de su utilización. Esto evita precargar diseños de casas que no van a ser visualizados en detalle por el usuario final.*

## *Patron Model View Presenter (Modelo vista presentador)*

*Este patrón lo utilizaremos porque nos ayuda a tener separado la vista de la lógica de negocio que en nuestra aplicación es importante ya que esto nos brinda pode hacer mantenimiento del software para futuras mejoras*

# Quality

*Los atributos de calidad que se necesitan para tener un mejor desempeño y cumplir con las exigencias que se tiene son:*

* ***Disponibilidad:*** *La idea que tiene la empresa con este tipo de página web, es fidelizar a los clientes, atraer clientes pero sobre todo tener sus datos para posteriores contactos e incluso promociones. Por este motivo se necesita tener alta disponibilidad en el servicio ya que si esto no se cumple causaría demoras en cotizaciones y no ayudaría al objetivo.*

*Para garantizar la disponibilidad en la página web se utilizara una técnica llamaba balanceador de tráfico la cual aumenta la disponibilidad del servicio ya que si un servidor falla, el servicio no se verá afectado. Por lo tanto requiere un servicio web de alta disponibilidad se deben utilizar dos balanceadores de tráfico.*

***Balanceo de trafico:*** El balanceo de tráfico es una técnica para distribuir la carga de forma equitativa entre dos o más servidores de forma que se obtenga una utilización de los recursos óptima, maximice el rendimiento, minimice el tiempo de respuesta y se evite la sobrecarga. El servicio de balanceo de tráfico se ofrece a través de un programa destinado a tal efecto o bien a través de hardware dedicado.

* ***Escalabilidad:*** *La escalabilidad es otro de los criterios de calidad que se desean en el proyecto, ya que el proyecto está iniciando y va a tener constante crecimiento. Por lo tanto se va a escalar de forma horizontal, ya que escalar de forma vertical tendrían unos costos mucho más elevados porque se necesitaría un hardware de alto rendimiento.*

*Al escalar de forma horizontal, las páginas web que contienen la información deben ser replicadas a todos los servidores web. Para poder obtener mejoras después de añadir más servidores, las peticiones de los distintos clientes se deben repartir de forma equilibrada entre los servidores. Esta tarea de redirigir peticiones hace necesario utilizar un balanceador de tráfico web que se encargue de redirigir el tráfico hacia cada uno de los servidores.*

*En un entorno web escalado de forma horizontal se debe tener muy en cuenta el tratamiento y persistencia de las sesiones. Tener sesiones  almacenadas de forma local en cada servidor es una mala práctica. La solución pasa por tener sesiones centralizadas o, incluso mejor, no tener  sesiones en absoluto.*

### **Arquitectura shared nothing:** Para que las aplicaciones puedan escalar horizontalmente se suele utilizar una arquitectura shared nothing. En una arquitectura shared nothing cada uno de los nodos de la aplicación son independientes, auto contenidos y autosuficientes. Como cada nodo puede trabajar por sí solo sin depender de sistemas externos, al aumentar la demanda se pueden añadir tantos nodos nuevos como sean necesarios para cubrirla. Gracias a los servicios en la nube como Amazon EC2 o el Google App Engine el coste de añadir nuevos nodos es cada vez más bajo. Simplemente se crea una imagen del disco duro del sistema con la aplicación y se programa para que arranque automáticamente. Si la afluencia de público aumenta considerablemente se pueden arrancar nuevas instancias para cubrir la demanda.