PUERTA DE ACERO

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 26/mayo/19 | 1.1 | Ultima revisión | Luis García |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Architectural Representation 3

3. Architectural Goals and Constraints 3

4. Use-Case View 3

4.1 Use-Case Realizations 3

5. Logical View 3

5.1 Overview 3

5.2 Architecturally Significant Design Packages 3

6. Process View 3

7. Deployment View 3

8. Implementation View 3

8.1 Overview 3

8.2 Layers 3

9. Data View (optional) 3

10. Size and Performance 3

11. Quality 3

# Introduction

En el presente document se muestra la arquitectura que tendrá el Sistema Puerta de acero que será desarrollado por nuestra compañía

## Purpose

Acontinuación se llevará a cabo una descripción de la arquitectura que tendrá nuestro sistema

## Scope

Se pretende que con este document se tenga una vision clara de la arquitectura que tendrá el Sistema y todos sus componentes

## Definitions, Acronyms, and Abbreviations

## References

## Overview

# Architectural Representation

[Esta sección describe qué arquitectura de software es para el sistema actual y cómo se representa. De las Vistas de Caso de uso, Lógica, Proceso, Implementación e Implementación, enumera las vistas que son necesarias y, para cada vista, explica qué tipos de elementos de modelo contiene.]

El Desarrollo de nuestro Sistema tendrá una arquitectura de Desarrollo llamada MVC o desglosado, Modelo, Vista, Controlador, con la cual se pretende que el Desarrollo del mismo se torne más fácil para los programadores, ya que es un patron de Desarrollo conocido y utilizado por muchos, además de tener la ventaja de que es de una mantenibilidad muy buena.

# Architectural Goals and Constraints

Debido a que es un nuevo desarollo no se está heredando Código, sin embargo si se utilizará un framework llamado Materialize, el cual facilitará en gran manera el diseño o front end ya que para dicho propósito está hecho.

Las herramientas de Desarrollo que se utilizarán son:

* Sublime text como editor de Código
* XAMPP como servidor web y de base de datos, el cual contiene PHP y MySQL indispensables para el funcionamiento del Sistema

# Use-Case View

En el caso de nuestro Sistema, se hace mención al caso de uso de login, el cual es de suma importancia para mantener la seguridad del Sistema, ya que de la correcta autenticación de los usuarios depende el ingreso al Sistema y en consecuencia del acceso a los datos importantes para nuestro cliente por lo cual se hace énfasis en que el login debe contener la mayor complejidad en desarollo para mantener la seguridad.

## Use-Case Realizations

Cuando el usuario pretenda ingresar al Sistema, el login será el encargado de mantener la seguridad del Sistema.

Al autenticarse correctamente, el usuario, si es administrador podrá acceder a los menús:

* Usuarios
* Casas
* Pagos

Si el usuario es operative solo podrá acceder al menu:

* Pagos

# Logical View

El Sistema contará con las clases:

* Usuarios
* Casas
* Pagos
* Conexión

Ya que con la division del Sistema en estas clases, se logra la total abstracción del problema y se logrará resolver cada una de las secciones necesarias para dicho Sistema son necesarias para cubrir los requerimientos dados por el cliente y se logra una modularización, para poder trabajar en diferentes equipos y darle agilidad al proceso tanto de Desarrollo como de pruebas.

## Overview

## Architecturally Significant Design Packages

Una de las clases relevantes del Sistema es la conexión, ya que será la encargada de dar la funcionalidad y comunicación con la base de datos, hecho que es indispensable para el funcionamiento del Sistema en cuestión

# Process View

# Deployment View

# Implementation View

## Overview

## Layers

# Data View (optional)

# Size and Performance

# Quality