Instituto Tecnológico de Culiacán

Ingeniería en Sistemas Computacionales

Proyecto: Almacén de partes de equipo de cómputo

Profesor: Carlos Santillan

Participantes:

Juan Pablo Bracamontes Astorga

Héctor Eduardo García Mata

Jorge Luis Lopez Payán

Victor Alain Torrecillas Camacho

Manuel Alejandro Verdugo Perez

Alejandro de Jesus Zazueta Peñuelas

|  |  |  |
| --- | --- | --- |
| Capítulo | Tema | Página |
| 1 | Prefacio |  |
| 2 | Introducción |  |
| 3 | Requisitos del sistema |  |
| 3.1 | Funcionales |  |
| 3.2 | No Funcionales |  |
| 4 | Arquitectura del sistema |  |
| 4.1 | Definir la arquitectura |  |
| 4.2 | ¿Qué es la arquitectura cliente-servidor? |  |
| 4.3 | ¿Por qué se eligió esta arquitectura? |  |
| 5 | Actores y roles |  |
| 6 | Casos de uso |  |
| 6.1 | Diagramas |  |
| 6.2 | Especificaciones |  |
| 7 | Modelo de Objetos |  |
| 8 | Diagramas de secuencia |  |
| 9 | Diagramas de clases |  |
| 10 | Contratos |  |

1.- Prefacio

Este documento de requisitos del proyecto de un sistema web para un almacén de partes de equipo de cómputo, el cual tiene como propósito el control sobre la salida de partes de equipo de cómputo.

Alcance del documento:

El documento de requisistos será la base de todo nuestro desarrollo futuro de nuestro almacén de equipo de cómputo. Describe los siguientes aspectos de nuestro sistema:

Propósito, requisitos funcionales y no funcionales, requisitos de las pruebas, arquitectura del sistema y los riesgos del proyecto.

Documentos relacionados:

Documentos de inicio del proyecto del almacén de equipo de cómputo.

Autores:

Juan Pablo Bracamontes Astorga, Héctor Eduardo García Mata,

Jorge Luis Lopez Payán, Victor Alain Torrecillas Camacho,

Manuel Alejandro Verdugo Perez, Alejandro de Jesus Zazueta Peñuelas

Lectores:

El documento esta dirigido principalmente a los desarrolladores del sistema web para un almacén de partes de equipo de cómputo, como al mismo tiempo sirve como base para todos aquellos que quieran realizar un proyecto de este tipo.

2.- Introducción

Se describe brevemente el contexto, los objetivos y el alcance del proyecto que estaremos desarrollando, así como la documentación del mismo.

2.1.- Propósito: El desarrollo del proyecto busca llevar un mejor control sobre la salida de producto y para que todo el proceso se realice de forma automática y electrónica.

2.2.- Alcance: El proyecto esta dirigido para una empresa que no cuenta con un servicio web para la salida de partes de cómputo, hasta llegar a la fase de implementación del mismo.

2.3.- Contexto: Nuestro proyecto se esta elaborando en el Framework Laravel 5, con el editor de texto Sublime Text 3 y con las herramientas para trabajar remotamente llamadas Git Hub y Git Bash, que esto nos permite llevar un control de versiones de nuestro proyecto.

3.- Requisitos del Sistema

3.1.- Requsistos Funcionales

RF1.- La aplicación debe poder permitir dar de alta nuevos partes de equipo de cómputo a través de la página web.

RF2.- La aplicación debe poder permitir la creación de un PDF con el folio, el usuario, la fecha y los productos que saco del almacén.

RF3.- La aplicación debe de mostrar las partes de cómputo filtradas (Fuente de poder, Tarjeta Madre, Disco Duro, etc.).

RF4.-

3.2.- Requisitos no Funcionales

RNF1.- Se debe poder ingresar la aplicación en cualquier momento.

RNF2.- Los datos de los usuarios deben de estar protegidos en todo momento.

RNF3.- Se debe poder ingresar rápidamente a la aplicación.

RNF4.- La interfaz de la aplicación debe de ser fácil de usar.

4.- Arquitectura del Sistema

4.1.- Definir el tipo de Arquitectura

La arquitectura que creemos que se adapta mejor a nuestro sistema: es la arquitectura de cliente-servidor, con patrón de vista controlador.

4.2.- ¿Qué es la arquitectura cliente-servidor?

Es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.



Modelo Vista Controlador

 Es un patrón de arquitectura de software 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. Para ello, el MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.



4.3.- ¿Por qué se eligio esta arquitectura?

Se escogió la arquitectura cliente-servidor por que se nos permite un mejor aprovechamiento de los recursos de cómputo disponibles. Una buena implmentación de la misma arquitectura, nos reducirá de manera importante el tráfico de datos a traves de nuestra red. También nos permite el uso de interfaces gráficas de fácil entendimiento para el usuario final, este tipo de arquitectura facilita la integración entre diferentes sistemas.