ARTIFACT 4

ALEX JOSÉ ALBERTO BARRETO

CRISTIAN DAVID GONZÁLEZ CARRILLO

NICOLÁS RESTREPO TORRES

JULIÁN ESTEBAN SALOMÓN TORRES

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE BOGOTÁ

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS E INDUSTRIAL

INGENIERÍA DE SOFTWARE I

BOGOTÁ D.C.

2016

ARTIFACT 4

Dentro de las modelos de arquitectura estándar nuestro software maneja una estructura estática que no cambia en el tiempo de ejecución sino que más bien se planea y se diseña con base a las necesidades específicas del negocio y una arquitectura orientada a objetos, en donde cada cosa representa un objeto que puede ser manipulado por el sistema como un todo, la construcción se trabaja con base al patrón MVC, donde se busca colocar en marcha el programa con tres capas distintas, precisamente el modelo, la vista y el controlador con el fin de desacoplar lo mayor posible todas las mismas y cohesionarlas para que cada una se encargue de la parte del código que tienen como objetivo con ideas básicas como la reutilización de códigos y separación de conceptos. Estas características tienen como objetivo facilitar todas las etapas del ciclo de desarrollo de software dentro de las posibles aplicaciones del software final a su destinación final como lo es la odontología.

El modelo representa la construcción de información con la que operará el sistema, dentro de las capas se encontrarán mecanismos que permiten interactuar con información que se encuentra previamente almacenada mediante funciones de selección, actualización, inserciones y eliminación de datos así como otras que se pueden cargar a determinadas partes de la aplicación web. En la construcción de Dentistware se modelan las distintas partes del software como modelos que se cargan para acceder a los datos.

La vista nos permite generar un encuentro de la interfaz, la cual permite interactuar con la aplicación, y visualizar donde usualmente no se tiene lógica alguna de trabajo, por ejemplo en cuanto al manejo específico de información, se cuenta con funciones que permiten renderizar los estados de dicha aplicación mediante código HTML5, CSS3 y PHP para visualizar una salida agradable al usuario, aquí se ve la alta cohesión desde que existe una vista “maestra” que contiene todos los elementos básicos del código HTML5, esto ayuda a simplificar el trabajo además de proveer una retícula uniforme y una paleta colores agradable para el usuario final.

El controlador es donde se relaciona la vista y el modelo que representa la información de nuestros usuarios, la utilidad de dicho código representado en este punto es servir de enlace entre la manipulación de datos, responder acciones o eventos generados en la aplicación y las salidas presentes en la interfaz, también tiene métodos que responden a ciertos inputs dentro del sistema que de forma asíncrona funcionan para entregar información inmediata al usuario o para hacer cambios en la información mediante un modelo, esto mediante AJAX que no es más que una técnica de desarrollo para hacer este tipo de acciones basada en JQuery. Dentro de las otras tecnologías con que cuenta el proyecto usamos Bootstrap para manejar todas las interfaces y conseguir la uniformidad deseada que se mencionaba anteriorimente, para hacer los llamados desde el modelo y manejar la base de datos hecha en MySQL se usa un framework full-stack que orquesta las tres capas básicas y crea una forma organizada de trabajo, este es Codeigniter una implementación de MVC en lenguaje PHP.

**REFERENCIAS BIBLIOGRÁFICAS.**

Wikipedia (octubre de 2016), (23 de octubre de 2016). Modelo–vista–controlador. Recuperado de: https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador

desarrolloweb.com (de enero de 2014), (23 de octubre de 2016). ¿Qué es MVC? Recuperado de:

http://www.desarrolloweb.com/articulos/que-es-mvc.html.