**Laboratorio 2**

****

**Ingeniería de Software II**

**Presentado por:**

**Jonathan Felipe Hurtado Diaz**

**Juan Carlos Fernandez Cuetia**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Programa de Ingeniería de Sistemas**

**Popayán, Cauca**

**2023**

**OBJETIVO:**

**Analizar diferentes estilos de arquitecturas de software, y**

**desarrollar criterios para seleccionar la arquitectura apropiada para un sistema.**

**CONSULTAR:**

**1. ¿Qué es una arquitectura de software?**

Se refiere a una planificación basada en modelos, patrones y abstracciones teóricas al momento de realizar una pieza de software y previa a cualquier tipo de implementación. También puede entenderse como una serie de patrones o lineamientos que se deben seguir en la construcción de un programa. Estos lineamientos ayudan a los desarrolladores, analistas y demás cargos relacionados a lograr cumplir con la construcción y desarrollo de un software.

La manera en la que se diseña o estructura un sistema tendrá un gran impacto en la capacidad de este en satisfacer los atributos de calidad, como el desempeño, la usabilidad o la modificabilidad. Así mismo, los diseños arquitectónicos pueden ser reutilizados para construir otros sistemas, permitiendo reducir costos y tiempos de planeación y desarrollo.

**2. ¿Cuál es el papel de un arquitecto de software?**

El arquitecto de software es la persona encargada de implementar, diseñar y ejecutar soluciones de software según las necesidades específicas de una empresa o cliente, aplicando patrones o modelos de arquitectura de software con la finalidad de construir un software limpio y seguro. Este además de diseñar el sistema basado en los requerimientos también debe realizar un seguimiento de la arquitectura tras la implementación del sistema con el fin de asegurar que se cumplan satisfactoriamente las necesidades pedidas por la empresa o el cliente.

**3. Seleccione dentro de la siguiente lista de estilos arquitectónicos dos para su estudio en profundidad:**

**Tuberías y filtros**

**Pizarra**

**Capas**

**Orientados a Objetos**

**REST**

**Máquina Virtual**

**Modelo Vista controlador**

**Analice sus características, principales ventajas, desventajas y sistemas en los cuales es recomendable su uso. Construya una tabla comparativa.**

| **Modelos** | **MVC** | **REST** |
| --- | --- | --- |
| **Objetivo** | Separar los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. | Proporcionar una interfaz uniforme para realizar operaciones en los recursos de la red. |
| **Ventajas** | Separación de preocupaciones, lo que facilita la administración y el mantenimiento del código. | Simplicidad y flexibilidad, ya que utiliza el protocolo HTTP y puede ser utilizado con prácticamente cualquier lenguaje de programación y plataforma. |
| **Desventajas** | Conexión demasiado estrecha entre la vista y el controlador, acceso ineficiente a los datos del modelo y aumento de la complejidad de la estructura y la implementación del sistema. | Puede ser difícil diseñar una API RESTful que cumpla con todos los criterios de REST, y el uso de verbos HTTP puede limitar la expresividad de las operaciones en los recursos. |
| **Sistemas de Aplicación** | Aplicaciones web y móviles, sistemas empresariales, etc. | APIs para aplicaciones web y móviles, integración de sistemas empresariales, conexión de dispositivos IoT, etc. |

**4. Recientemente ha habido muchas inconformidades con el censo en Colombia que se realizó en 2018, el gobierno ha decidido estudiar la posibilidad de realizar uno nuevo para el año 2022 y así corregir los errores del anterior para tener unas cifras más reales, las cuales además han cambiado dado el fenómeno migratorio venezolano. Su empresa ha ganado la licitación para este nuevo proyecto, y una de sus primeras decisiones será la de seleccionar una arquitectura adecuada para el nuevo sistema.**

**¿Considera que alguna de las arquitecturas estudiadas sería útil para esta tarea?, ¿por qué sí?, ¿por qué no?**

**Para ello revise bien que es un censo, ¿cómo se realiza?, ¿qué tipo de información se colecta?, ¿qué información debe arrojar?, ¿qué componentes debe tener un sistema que lo soporte, igualmente revise que atributos de calidad debe tener.**

**Recuerde somos más de 50 millones de Colombianos hoy (50.270.571).**

Consideramos que el patrón de arquitectura Modelo - Vista - Controlador (MVC) podría sernos útil para llevar un el proyecto de un censo ya que nos brinda diversas funcionalidades tales como:

1. Separación de responsabilidades: MVC separa la lógica de negocio (Modelo), la interfaz de usuario (Vista) y las acciones basadas en la entrada del usuario (Controlador). Esto facilita la organización del código y hace que el sistema sea más manejable.

2. Reutilización de código y modularidad: Los componentes en MVC son independientes, lo que permite reutilizar el código y mejorar la modularidad.

3. Paralelismo en el desarrollo: Dado que los componentes son independientes, diferentes equipos pueden trabajar en el Modelo, la Vista y el Controlador simultáneamente.

4. Adaptabilidad: Si necesitas cambiar la interfaz de usuario o la forma en que se accede a los datos, puedes hacerlo sin tener que modificar todo el sistema.

Un censo es un recuento de los individuos que conforman una población estadística. Se realiza para obtener información estadística relevante y suele abarcar un período determinado. Los datos que suelen recogerse en un censo son: nombres, tipo de identificacion, identificacion, edad, sexo, nivel de educación, acceso a servicios básicos, ingresos mensuales, características de la vivienda habitada, entre otros.

Un sistema que soporte un censo debería tener los siguientes componentes:

1. Interfaz de usuario: Para recoger los datos del censo. Esto podría ser una aplicación web o móvil donde los usuarios pueden ingresar sus datos.

2. Base de datos: Para almacenar los datos recogidos.

3. Servidor: Para gestionar las solicitudes de los usuarios e interactuar con la base de datos.

4. Sistema de autenticación: Para verificar la identidad de los usuarios y proteger sus datos.

5. Sistema de informes: Para generar informes basados en los datos recogidos.

En cuanto a los atributos de calidad, un sistema de censo debe ser:

1. Seguro: Los datos personales deben estar protegidos.

2. Escalable: El sistema debe ser capaz de manejar un gran número de usuarios.

3. Fiable: El sistema debe funcionar correctamente y sin errores.

4. Fácil de usar: La interfaz de usuario debe ser intuitiva.

5. Accesible: El sistema debe ser accesible para todos los usuarios, independientemente de su habilidad o tecnología.

Es por estas razones que escogimos el patrón de arquitectura (MVC) ya que dentro de todos los patrones de arquitectura estudiados es aquel que nos brinda una mayor ayuda, fiabilidad y sencillez a la hora de llevar a cabo un proyecto tan grande y con tantos datos como lo es un censo.

**5. Prepare un informe de esta práctica y entréguelo en las mismas condiciones que el laboratorio anterior.**