

Actividad foro decisiones de diseño

Laura Sofía Dueñas Bulla - 20172020102

Daniel Camilo Carreño Merino - 20181020148

Jhoan Sebastian Wilches Jimenez - 20181020175

Pablo Esteban Espinosa Granados - 2018102001

Julieth Tatiana Riaño Mora - 20181020107

Febrero 2023

Universidad Distrital Francisco José de Caldas

Facultad de Ingeniería

Seminario de ingeniería

Acta de reunión 22/02/2023

Modelamiento para la arquitectura del sistema de información *CONSORCIO COMIDAS A DOMICILIO*.

Se realiza la correspondiente reunión del equipo: “Los Más Melos” en la fecha estipulada, en donde se llevó a cabo un debate del listado de modelos a desarrollar y su importancia para la arquitectura y desarrollo del proyecto.

Debido al tipo de metodología de desarrollo elegida por el equipo, el proceso puede pasar por una serie de cambios y modificaciones se necesita establecer y documentar de forma clara la arquitectura de diseño para tener un flujo de desarrollo que tienda a minimizar los riesgos.

Los modelos de desarrollo de software son herramientas fundamentales para garantizar que los sistemas de información se construyan de manera efectiva y eficiente. Estos modelos proporcionan un marco estructurado y sistemático para diseñar, desarrollar y mantener software de calidad.

Al utilizar un modelo de desarrollo de software, los equipos de desarrollo pueden disfrutar de varios beneficios. En primer lugar, los modelos establecen prácticas y pasos que ayudan a mejorar la eficiencia y productividad del equipo. Además, se incluyen prácticas de aseguramiento de calidad que garantizan que el software final cumpla con los estándares y requisitos establecidos.

Otro beneficio que vale la pena mencionar es la mejora en la gestión de riesgos y el control de cambios en el proyecto. Al tener una metodología estructurada, el equipo de desarrollo puede tener un mejor control sobre el proyecto y su evolución.

Por otro lado, los modelos de desarrollo de software fomentan la comunicación y colaboración entre los miembros del equipo de desarrollo y los stakeholders del proyecto. Al seguir una metodología sistemática, se pueden evitar malentendidos y errores de comunicación.

Finalmente, los modelos de desarrollo de software son altamente adaptables y personalizables. Lo cual permite a los equipos de desarrollo adaptar el modelo a las necesidades específicas del proyecto y hacer ajustes según sea necesario.

Teniendo en cuenta lo anterior los modelos que se proponen son los siguientes:

Modelo relacional

Este modelo nos ayudará a definir un marco de relaciones entre datos dentro de la base de datos, así como una guía para el uso de los mismos. Los modelos de datos son un elemento fundamental para el desarrollo de este software y sus requerimientos funcionales y no funcionales. Nos permite un método estandarizado para definir y formatear los contenidos de la base de datos de forma consistente en todos los sistemas, lo cual permite que diferentes requerimientos compartan los mismos datos.

Modelo de estados

Para el caso de este proyecto, el modelo es necesario para representar el estado de un pedido, por ejemplo, iniciado, pendiente, finalizado y/o cancelado. Las acciones que realice el usuario mueven el estado del reporte hacia el siguiente. Asimismo, el domiciliario tendrá diferentes estados dependiendo de si tiene actualmente un pedido asignado, si se encuentra a la espera de pedidos o si no se encuentra disponible en el momento.

Diagrama de despliegue

Debido a que se planea que este sistema cambiará sus estados por variables fuera del software, es necesario el diagrama de despliegue para visualizar los distintos nodos y/o dispositivos de hardware del sistema, los enlaces de comunicación entre los distintos restaurantes que se muestran en la aplicación, involucrados y la colocación de los archivos de software en ese hardware. Así como las diferentes aplicaciones de control (software de clientes, software de domiciliarios o software de restaurante).

Diagrama de paquetes

Este diagrama nos ayuda a simplificar los diagramas de clase debido a que pueden agrupar dichas clases en paquetes, haciendo más comprensible las relaciones entre los diferentes actores y componentes del sistema (restaurantes, pedidos, clientes, comidas e ingredientes).

Diagrama de componentes

El diagrama de componentes nos brinda una visión general del sistema y documenta la estructura de los componentes en el sistema, sus relaciones y dependencias mutuas, además le dan al desarrollador la información necesaria sobre si un sistema funciona de forma coherente, cumple sus tareas y/o objetivos del mismo.

Diagrama de casos de uso

El diagrama de casos de uso es una forma de diagrama de comportamiento en lenguaje de modelado unificado (UML), con la que se representan procesos empresariales, así como sistemas y procesos de programación orientada a objetos. En otras palabras, este diagrama permite observar las funcionalidades del sistema desde el punto de vista de un actor, este actor puede ser un usuario o un sistema que accede a otro, con la finalidad de que se observe la relación entre un actor y sus requisitos o expectativas del sistema.

Siendo las 13:06 del día fechado se da por terminada la reunión con un acuerdo por todas las partes para el tema: modelamiento para la arquitectura de software.

Firman:

Líder:

Laura Sofía Dueñas Bulla

20172020102

Líder de calidad:

Pablo Esteban Espinosa Granados

2018102001

Líder de planeación:

Jhoan Sebastian Wilches Jimenez

20181020175

Líder de soporte:

Daniel Camilo Carreño Merino

20181020148

Líder de desarrollo:

Julieth Tatiana Riaño Mora

20181020107