

WIS Architecture report

21-11-2022

Noviembre D05 – Diseño y Pruebas II

Grupo D07

CERRATO SÁNCHEZ, LUIS (luicersan@alum.us.es)

GUITÉRREZ CONTRERAS, ERNESTO (erngutcon@alum.us.es)

Tutor

PATRICIA JIMÉNEZ AGUIRRE

<https://github.com/erngutcon/Acme-Courses-D05>

Índice

Índice	1
Resumen	2
Tabla de revisión	3
Introducción	4
Contenido	5
Conclusiones	7
Bibliografía	8

Resumen

En este documento podemos encontrar una introducción sobre que representa el informe de Sistemas de Información Web y se muestra un contenido sobre los conocimientos y los conceptos que hemos obtenido respecto a la arquitectura de Sistemas de Información Web durante la realización de los entregables de la asignatura Diseño y Pruebas II.

Tabla de revisión

Número de revisión	Fecha	Descripción
V1.0	21-11-2022	Crear documento y recopilar información.

Introducción

Tal y como se describe en el resumen, este documento se centra en los conocimientos obtenidos sobre los sistemas de información web en la asignatura Diseño y Pruebas II.

A lo largo del documento, describiremos lo que hemos aprendido en esta asignatura sobre la arquitectura de sistemas de información web, tanto lo nuevo, como lo que ya conocíamos de la asignatura de Diseño y Pruebas I y hemos ampliado este cuatrimestre.

Los sistemas de información que usan tecnologías web pueden integrar procesos o sistemas dentro de una única interfaz y permitir el acceso a través de una intranet local o una red de Internet global y son sistemas que las distintas organizaciones y sus clientes utilizan para realizar por ejemplo el comercio electrónico.

A fecha de finalización de este documento, hemos aprendido bastante más al respecto del funcionamiento, los elementos que la componen y la estructura de los sistemas de información web.

Contenido

Un sistema de información web es un sistema de información que utiliza tecnologías web de Internet para ofrecer información y servicios a usuarios u otros sistemas o aplicaciones de información. Su objetivo principal es publicar y mantener datos utilizando principios basados en hipertexto y por lo tanto podríamos decir que es un "esqueleto" o diseño que muestra las interacciones entre los componentes de la aplicación, los sistemas de middleware, las interfaces de usuario y las bases de datos.

La arquitectura de los sistemas de información web tiene tres componentes principales:

- Un navegador web o cliente que es la capa del navegador.
- Un servidor de aplicaciones web que se representa por la capa de aplicación.
- Un servidor de base de datos que se representa mediante la capa de la base de datos.

Las aplicaciones de bases de datos se basan en un servidor de bases de datos, que proporciona los datos para la aplicación. El servidor de la base de datos ofrece lógica comercial e información o datos relevantes que el servidor almacena y administra, recupera y proporciona la información.

Durante los cursos en la Universidad de Sevilla nos hemos enriquecido de conocimiento sobre por ejemplo cómo utilizar y aplicar un diseño arquitectónico de una aplicación Spring Boot con Spring MVC que es el patrón modelo-vista-controlador y se trata de un patrón arquitectónico que estructura una aplicación en 3 componentes como el modelo que es la representación de la información, incluye los datos y la lógica de negocio necesaria para trabajar con ellos. La lógica de negocio no se debe colocar fuera del modelo, la vista que es la representación del modelo de manera que el usuario puede interactuar con él y suele hacerse mediante una interfaz de usuario y el controlador que responde a eventos en la interfaz de usuario (solicitudes HTTP en nuestro caso), invoca cambios en el modelo y probablemente en la vista.

Hemos aprendido en utilizar los controladores que son las clases/funciones que maneja todas las solicitudes de un sitio web y luego las envía al controlador apropiado, en utilizar la base de datos llamada MariaDB y junta con el gestor DBeaver nos permite tener acceso en tiempo real a nuestros datos y también utilizar los servicios en los cuales están basados los repositorios que están interactuando con las bases de datos y dependiendo de donde queremos mostrar los datos tenemos que utilizar el Show de AbstractService o el List de la misma interfaz. En nuestro proyecto mediante los controladores que los tenemos de AbstractController presentan 2 parámetros los roles y los objetos. Los controladores responden a eventos en la interfaz de usuario (HTTP request en nuestro caso), invoca cambios en el modelo y probablemente en la vista. Spring da una manera de mapear las HTTP request como métodos con @GetMapping o @PostMapping Los controladores son responsables de transformar entidades en un ModelMap que puede ser procesado por las vistas, validación de la entrada proporcionada por la vista, transformar esta entrada en entidades si es necesario, llamar a servicios para realizar una acción y devolver el nombre de la vista que se debe cargar. Para conectar el controlador con el modelo solo necesitamos inyectar la dependencia de los servicios en el controlador y también hay que nombrar que para trabajar con los flujos de trabajo hemos necesitado de la

implementación de algunos comandos como update y publish, list y show, create y delete para añadir o eliminar, representar o modificar entrada en la base de datos.

Como ya sabemos la arquitectura de información web admite la lógica comercial y las aplicaciones de múltiples capas, desarrolladas con Python, PHP, Java, .NET, Ruby y Node.js.

También hemos aprendido en utilizar las vistas que representan la información dada por el controlador para especificar cómo se debe representar una interfaz de usuario para una función determinada y en nuestro proyecto están representadas por los ficheros jsp y las solicitudes a un sistema de información web se realizan mediante solicitudes HTTP a una URL en particular.

Nos hemos confrontado con los patrones de la capa lógica de negocios como transaction script donde se organiza la lógica empresarial por procedimientos, donde cada procedimiento maneja una sola solicitud de la presentación., domain model que es un modelo de objeto del dominio que incorpora ambos comportamientos y datos y service layer que define el límite de una aplicación con una capa de servicios que establece un conjunto de operaciones disponibles y coordina la respuesta de la aplicación en cada operación, la capa de presentación interactúa con el dominio a través de la capa de servicio y la capa de servicios divide la lógica empresarial en dos tipos diferentes como la lógica de dominio y la lógica de aplicación.

La arquitectura del sistema de información web Java se basa en que es un lenguaje de programación versátil y es popular en el entorno de desarrollo empresarial. Independientemente de la complejidad o la naturaleza de la aplicación, la arquitectura de aplicaciones web de Java es la plataforma preferida por los desarrolladores para crear soluciones y cumplir con las expectativas. Una de las ventajas distintivas de esta arquitectura es la capacidad de combinar y confiar en las herramientas nativas de Java y los marcos para crear aplicaciones. También hemos aprendido a utilizar la vinculación de parámetros HTTP y objetos Java y cómo podemos llenar un objeto java usando los valores de un formulario HTML, que es un caso de uso muy común, necesitamos ir mapeándolos manualmente uno por uno y por ejemplo en Sprint se usa el concepto de binding, que vincula automáticamente un conjunto de parámetros HTTP a los atributos del objeto Java.

Para nombrar algunos ejemplos de sistemas de información que son utilizados por las distintas empresas como los siguientes: sistema de soporte de decisiones, sistema de control de procesos, sistema de Información de Gestión, almacenes de datos, planificación de recursos empresariales, etc.

Conclusiones

Gracias a todos los conocimientos adquiridos mencionados, pensamos que poseemos nuevas aptitudes y un nuevo punto de vista sobre el funcionamiento de los distintos aspectos de la arquitectura de un sistema de información web.

Por lo tanto este aprendizaje será de gran ayuda y utilidad para nosotros como estudiantes e intentamos prepararnos para desempeñar un papel de liderazgo en la ciencia y la tecnología de los sistemas de información web, desarrollar una comprensión profunda de la construcción, métodos, técnicas y herramientas para el uso de dicho sistema.

Hemos e intentamos cada día esforzarnos por avanzar en las distintas tareas como el modelado de usuarios, la recuperación de información, el procesamiento del lenguaje natural, los sistemas de bases de datos, la ingeniería web, la gestión de datos web, la interacción del usuario, etc.

Bibliografía

Intencionalmente en blanco