

DIAPPOSITIVA Nº2

- Cada vez más gente tiene horario de oficinas y necesitan comer en el trabajo, o bien porque no les da tiempo de ir a casa o bien porque no tienen tiempo o habilidades para gestionar su menú semanal . Por lo que una herramienta de solicitud de menús les mejoraría su calidad de vida, que repercutiría en una mejora de su productividad.
- Se propone una solución: “Mi Menú de Hoy” un aplicativo web basada en arquitectura J2EE, es un estándar de mercado, seguro, maduro, multiplataforma, sin necesidad del uso de licencias .
- Se decide implementar la solución utilizando el patrón MVC, mediante el framework Spring MVC. Este patrón ofrece las siguientes ventajas:
 - Podrás dividir la lógica de negocio del diseño, haciendo tu proyecto más escalable.
 - Te facilitará el uso de URL amigables, importantes para el SEO
 - Puedes utilizar abstracción de datos, en este caso usaremos el framework **Hibernate**, facilitando la realización de consultas a la base de datos.
 - Dan facilidades para el control de la memoria Caché
 - En el caso de proyectos donde hay varios desarrolladores, el seguir métodos comunes de programación, hace que el código sea más entendible entre estos, pudiendo uno continuar el trabajo de otro.
 - Los frameworks MVC están creados para facilitar el trabajo de los desarrolladores, encontrarás clases para controlar fechas, URL's, Webservices, etc. lo que tiene una gran ventaja en cuanto a productividad. Inicialmente como es lógico habrá una curva de aprendizaje, pero luego tendrás muchos beneficios.
 - Un Framework MVC te ayuda a controlar los recursos del servidor, evitando **Bugs** que puedan repercutir en el rendimiento, por ejemplo, muchas veces olvidamos cerrar conexiones a la base de datos, sobrecargando el servidor.

DIAPOSITIVA Nº3

El objetivo del trabajo consiste en realizar el Análisis, Diseño y Desarrollo del aplicativo Mi Menú de Hoy con tecnología J2EE utilizando para ello el patrón de arquitectura de software MVC. Se escoge el framework Spring MVC por el nicho de mercado que dispone y sus expectativas de futuro, así como una preferencia personal ya que hace unos 7 años tuve mi primera experiencia con Spring, pero desde hace unos 5 años no he vuelto a usar esta tecnología y me gustaría refrescarme un poco, para ver como ha madurado la tecnología y si se ha simplificado un poco todo el proceso de crear un aplicativo web.

Para todo el desarrollo se utilizarán tecnologías actuales y vigentes en el mercado, como son: Java 7 como lenguaje, para la capa de Modelo se utilizará como ORM la tecnología Hibernate, para la capa de vista se utilizará Apache Tiles, Spring Security como gestor de la seguridad y Spring MVC como nexos de unión entre todos ellos en la capa de controlador. Para la capa de vista también se utilizará HTML, Javascript, JQuery, JQueryUI.

DIAPOSITIVA Nº4

La idea es desarrollar un aplicativo web, con dos puntos de entrada, uno para el administrador /admin, donde éste podrá listar, modificar, y crear tanto Ingredientes, Platos, Recetas, Menús, Suscripciones. El usuario Nutricionista solo tendrá acceso a la parte de Menús en las que podrá asignar los menús a las suscripciones de los usuarios.

Por otra parte existirá una parte pública para todo el mundo donde se podrán buscar recetas por Nombre, o por preparación, se podrán consultar estas recetas, y los usuarios nuevos podrán suscribirse al servicio, seleccionando el tipo de menú, y el periodo de vigencia, y una vez suscritos los usuarios podrán modificar sus datos personales o bien consultar los menús que tienen esta semana y la semana que viene.

DIAPOSITIVA Nº5

Se utilizará la metodología Métrica 3 en cascada, por lo que el ciclo comenzará realizando el Análisis Funcional (requerimientos, redacción de casos de uso ..), posteriormente realizaremos el Diseño Técnico, definir el modelo de datos y el diagrama de clases principalmente, y posteriormente redactar los diagramas de estado y secuencia para acabar de definir el comportamiento.

Por último realizaremos el desarrollo, utilizando Java, y las tecnologías que he comentado anteriormente.

...

Ahora explicaré superficialmente la parte de Análisis Realizado...

DIAPOSITIVA Nº6 – ANÁLISIS ACTORES

Como he comentado antes, los actores principales del sistema son tres:

Administrador, Cliente, y Nutricionista. Aparte estaría el usuario anónimo, que sería cualquier usuario que se conecta a internet y visita nuestra web.

- Usuario Anónimo o Invitado, puede realizar las siguientes acciones ...
 - Consultar Recetas
 - Suscribirse
- Usuario Cliente, puede realizar las siguientes acciones...
 - Login como cliente
 - Ver sus menús asignados
 - Modificar su perfil
- Usuario Nutricionista, puede Asignar Menús
- Usuario Administrador gestiona toda la plataforma, Gestión Ingredientes, Platos, Menús, Recetas, Asignaciones, Destacar Recetas.

DIAPOSITIVA Nº7 – ARQUITECTURA

Como he comentado antes hemos escogido por Spring MVC como framework MVC de nuestra aplicación, aquí muestro la multitud de componentes que se pueden integrar con Spring, y que lo convierten en un framework muy polivalente, donde raramente tendremos que reinventar la rueda desarrollando cosas propias, sino que tendremos que incorporar módulos ya existentes, ... eso si, las dificultades las encontraremos a la hora de integrarlos y combinar versiones entre ellos.

DIAPOSITIVA Nº8 – ARQUITECTURA II

Siguiendo con la arquitectura aquí muestro la arquitectura de Spring MVC que gira entorno a un Servlet , el DispatcherServlet , que levantará el framework Spring que con sus archivos de configuración es capaz de instanciar los controladores, ViewResolvers, Servicios, DAOs , y después usando IOC es capaz de inyectar automáticamente unos componentes dentro de otros, todo a partir de configuración y creando un sistema muy desacoplado.

El ciclo de petición es el siguiente:

- El servlet recibe la petición del navegador
- El servlet consulta el HandlerMapping que es el encargado de traducir las urls en métodos dentro de controllers.
- Una vez obtenido el controller, se realiza la petición al mismo, el controller responde con un nombre de vista.
- El servlet vuelve a consultar el ViewResolver para determinar cómo mostrar la vista.
- Una vez el Dispatcher Servlet sabe que vista mostrar el Servlet contesta al navegador con la página solicitada.

DIAPOSITIVA Nº9 – ARQUITECTURA III

La aplicación se ha diseñado en base un modelo de 5 capas, la capa de controladores, nunca utilizará la capa de datos, utilizará una capa intermedia de servicios que dotará de mayor flexibilidad a nuestro desarrollo. Será esa capa de servicios la que trabaje directamente con la capa de datos, ésta a su vez, trabajará con la capa de persistencia implementada con Hibernate, por lo que el código resultante será bastante independiente a la Base de datos escogida.

DIAPOSITIVA Nº10 – ARQUITECTURA IV

La seguridad utilizaremos Spring Security que se integra perfectamente con SpringMVC mediante un fichero de configuración, se mapean urls a roles, y a unas tablas específicas dentro de la aplicación.

La persistencia como he comentado antes se realiza con HIBERNATE utilizando anotaciones que simplifican la configuración, y facilita el mantenimiento al realizar cambios en el esquema automáticamente.

Se utiliza un pool de conexiones configurable en el bean datasource para prevenir el sistema para un mayor volumen de datos y para evitar problemas de recursos con las conexiones a base de datos.

DIAPOSITIVA Nº11 – MODELO DE DATOS

Siguiendo con el diseño realizado, podemos ver en el aplicativo la relación principal, entre un Usuario que puede tener una o varias Suscripciones , en principio solo puede haber una activa en un momento dado, y esta suscripción tendrá asignado para cada día un menú, representado en Base de datos en la tabla Menus_Daily.

Un menú esta compuesto por tres platos que a su vez están compuestos de ingredientes.

Una receta es una implementación de un plato, o sea una manera de elaborarlo. Esto tuve dudas en cuanto a implementarlo o no porque un menú puede tener 3 recetas en lugar de 3 platos, pero al final lo implementé porque generalmente los menús de restaurante no son tan elaborados como las recetas de los chefs.

DIAPOSITIVA Nº12 – MMH – GESTIÓN

La parte de gestión se accederá mediante la url /admin y después de hacer Login, se nos mostrará un menú superior con las diferentes entidades a las que tenemos acceso, para cada una de ellas podremos ver el listado o bien crear una entidad nueva. En la pantalla de listado que aquí se muestra a la derecha tendremos opción de Editar o borrar el elemento de la fila. Si lo editamos , o creamos uno nuevo por menú vamos a la siguiente pantalla...

DIAPOSITIVA Nº13 – MMH – GESTIÓN

Donde en esta por ejemplo podemos rellenar los datos de una receta, en la parte inferior tendremos un botón de guardar y otro de cancelar. Con el botón de guardar crearemos o editaremos un nuevo registro y con el de cancelar volveremos al listado.

Al guardar las entidades generalmente se lanza una validación en servidor que nos retornará los errores sobre la misma página en letras en rojo.

También en la parte superior tendremos la opción de hacer logout.

DIAPOSITIVA Nº14 – MMH – PÚBLICA

La parte pública de la aplicación se accede con la url / , en esta no se nos pide inicialmente autenticación, y podremos buscar recetas, mediante el campo ubicado en la parte superior.

Tendremos también un menú, para acceder a las recetas, consultar el área privada, y podremos ver algunas recetas destacadas por el administrador.

Para suscribir un usuario accederemos a la opción de suscripción, se nos preguntará el periodo el tipo de menú, y nos preguntarán los datos del usuario.

Si clicamos en las recetas destacadas podremos ver la receta al completo con las imágenes y el resto de datos de la misma.