

evelb

PRUEBA TÉCNICA

EMILIO DOMINGUEZ SÁNCHEZ
emilio89@gmail.com

ÍNDICE

ÍNDICE	2
APLICACIÓN WEB.....	3
INTRODUCCIÓN	3
FUNCIONALIDES.....	3
ASPECTOS TÉCNICOS Y JUSTIFICACIÓN	4
IMPLEMENTACIÓN DE LA SOLUCIÓN Y JUSTIFICACIÓN	5
MANUAL DE INSTALACIÓN.....	7
Instalación y ejecución mediante Maven.....	7
Ejecución en Tomcat.....	7
IMAGENES DE LA APLICACIÓN.	8
INTEGRACIÓN CONTINUA.....	10
INTRODUCCIÓN	10
¿PORQUE LA INTEGRACIÓN CONTINUA?	10
SOFTWARE	10
FUNCIONAMIENTO	11
ACCESO A APLICACIÓN EN SERVIDOR, A JENKINS Y DIRECCIÓN GITHUB	11

APLICACIÓN WEB

INTRODUCCIÓN

Para la prueba técnica del proceso de selección que está llevando a cabo la empresa Evelb, se pide desarrollar una aplicación que nos permita conocer que tiempo hace en una localización (ciudad) concreta. Desde Evelb nos hacen llegar dos propuestas de implementación, una de ellas con interfaz gráfica y otra sin ella en la cual se mostrará el tiempo a través de la consola.

FUNCIONALIDADES.

Se implementa la primera opción de las propuestas por la empresa, la opción con interfaz gráfica.

Funcionalidades y datos mostrados:

- Nombre de la ciudad.
- Información adicional de la opción seleccionada. En este caso se mostrará también la humedad en la localidad seleccionada, la velocidad del viento y el nombre de cada estación meteorológica (en el propio mapa).
- Mapa centrado en la localización de la opción seleccionada con los puntos de información geográfica correspondientes a las coordenadas de las estaciones (y también su nombre en este caso).
- Una barra animada, que muestra la temperatura existente en ese momento en esa localización, con un tamaño y color que varíe en base a la temperatura. En este caso se considerará el siguiente rango de temperaturas y colores. Menos de 0° centígrados, color blanco. Entre 0° y 10° Centígrados color azul. Entre 10° y 25 ° color verde. Entre 25° y 35° color amarillo y finalmente más de 35° color Rojo.
- El usuario de la aplicación tendrá la posibilidad de volver a realizar otra búsqueda.
- Se incluyen logs por consola de las llamadas y respuestas realizadas a servicios

Funcionalidades extra:

- Aplicación multidioma. Se implementa una opción para poder cambiar el idioma de la aplicación (en este caso para castellano gallego). No así de los datos solicitados al servicio (geoname) que como es un servicio

de prueba solo se mostrarán en inglés, pero con esta opción y un servicio no de prueba sería fácilmente intercambiable también para que se pidiesen los datos en el idioma seleccionado.

ASPECTOS TÉCNICOS Y JUSTIFICACIÓN

Para esta aplicación se ha decidido usar una serie de frameworks que facilitan, agilizan y mejoran la tarea de desarrollo de la misma.

Spring. Es un framework para el desarrollo de aplicaciones y contenedor de inversión de control. Tiene una gran ventaja, ya que permite que los equipos de desarrollo puedan enfocarse directamente en la lógica empresarial que requiere la aplicación, haciendo el proceso más corto, rápido y eficaz, ahorrando líneas de código evitando tareas repetitivas. En este proyecto se ha elegido este framework puesto que (además de ser uno de los más usados en la actualidad) permite muy fácilmente el arranque de una aplicación como se comentará más adelante.

Spring Boot es una de las tecnologías dentro del mundo de Spring. Este nos permite desarrollar y arrancar de forma muy rápida aplicaciones basadas en Spring a través de la página <https://start.spring.io/> nos permite arrancar un proyecto muy rápido, ahorrandonos mucho tiempo en configuraciones, aspecto a tener en cuenta en este proyecto donde si bien no es crítico es importante entregar la aplicación cuanto antes.

Thymeleaf . es una librería Java que implementa un motor de plantillas de XML/XHTML/HTML5 (también extensible a otros formatos) que puede ser utilizado tanto en modo web como en otros entornos no web. Proporciona un módulo opcional para la integración con Spring MVC por lo que se hace ideal para este caso en concreto. Además se acopla muy bien para trabajar en la *capa vista* del MVC. El objetivo principal de Thymeleaf es permitir la creación de plantillas de una manera elegante y un código bien formateado. Se ha elegido esta librería ya que había leído mucho sobre el y sobre lo bien que se integraba con Spring y como tenía la posibilidad de un nuevo proyecto quería aprender a trabajar con este.

Junit .Es un framework que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase

cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba. Las pruebas unitarias de la aplicación se realizan con este framework puesto que es el más usado sin lugar a duda para estas tareas.

Bootstrap. Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Además soporta diseños web adaptables o responsivos. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles). Para el diseño de esta aplicación he elegido este framework puesto que como no soy un buen diseñador este framework facilita mucho la tarea. Además ya había trabajado en varios proyectos con el así que no sería inconveniente usarlo.

Jquery. Es una plataforma del lenguaje Javascript. Este ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Se elige esta biblioteca puesto que para la implementación de la barra de progreso de la temperatura se cree fundamental si no con Javascript llevaría mucho más código y tiempo implementarla. Además JQuery se acopla fácilmente y bien con Bootstrap.

Api Google Maps. Esta api es sólo HTML, CSS y JavaScript trabajando junto. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un <div> en la página HTML. Una de las ventajas de usar esta API es que es muy fácil de usar y dispone de una buena documentación, aunque sea una solución comercial como en este caso la aplicación será de prueba y no vamos a superar un número de visitas no necesitaremos servicios adicionales (que habría que pagar por ellos) en este caso con la versión de prueba nos llega.

IMPLEMENTACIÓN DE LA SOLUCIÓN Y JUSTIFICACIÓN

A continuación se detallarán aspectos de la implementación de la solución, comentando algunos aspectos técnicos y el porque de su elección.

Anotaciones de Spring usadas.

@SpringBootApplication – Se usa en la clase que inicializa la aplicación (TiempoApplication.java). Es una anotación que configura la aplicación web adicionando automáticamente todas las configuraciones de Spring y JEE que normalmente tendrían que hacerse manualmente. Esta anotación añade automáticamente todas las siguientes anotaciones:

- **@Configuration**. Marca esta clase indicando que hay **@Bean** que deben ser configurados al iniciar la aplicación.
- **@EnableAutoConfiguration** Le indica a spring-boot que comience a agregar los **@Bean** basándose en el classpath, configuraciones y otras propiedades.
- **@ComponentScan** El famoso y controversia esta pequeña anotación es la que nos libra para bien del xml, marca las clases que lo contienen (implícita o explícitamente) para ser escaneadas al iniciar la aplicación y configurar sus frijoles (**@Beans**)
- **@EnableWebMvc** esta habilita el viejo y conocido MVC de spring, lo que sucede mas bien es que dentro de nuestra dependencia spring-boot-starter-web se encuentra otra dependencia spring-webmvc lo cual hace que esta anotación sea implícitamente incluida aquí, como todo en spring puede o no usarse

@Controller: Anotación que registra el controlador para Spring MVC, es el que realiza las tareas de controlador y gestión de la comunicación entre el usuario y la aplicación .

@Service. Otra especialización de **@Component** (igual que **@Controller**) pero en este caso anota las clases a nivel de servicio. Se encarga de gestionar las operaciones de negocio más importantes a nivel de la aplicación y aglutina llamadas a varios repositorios de forma simultánea.

@Repository: Última especialización de **@Component** que se encarga de dar de alta un bean para que implemente el patrón repositorio que es el encargado de almacenar datos en una base de datos o repositorio de información que se necesite. En este caso aunque no tenemos una base de datos, se usa este repository para el acceso a las URLs de donde solicitamos los datos.

@RequestMapping: Anotación que se encarga de relacionar un método con una petición http. Se usa en los controladores. Con esta anotación de métodos definimos la URL en el navegador a la cual el controlador responderá

únicamente con las funciones GET y POST, el primero en el caso que el usuario consulta la pagina con un navegador web, el segundo cuando se envía información usando un formulario.

@Autowired. Esta anotación permite resolver mediante inyección las dependencias de un bean de Spring. Consiste que en lugar de que cada clase tenga que instanciar los objetos que necesite, sea Spring el que inyecte esos objetos, lo que quiere decir que es Spring el que creara los objetos y cuando una clase necesite usarlos se le pasaran (como cuando le pasas un parámetro a un método).

MANUAL DE INSTALACIÓN

Instalación y ejecución mediante Maven.

Es necesario tener una versión de Maven (a poder ser versión 3.3 o superior). Un JDK versión 8 correctamente instalado en el equipo. Si hay dudas con respecto a algún apartado, preguntar en el correo facilitado en este documento

Abrir una consola.

Ir al directorio del código fuente (facilitado junto a este documento).

Ejecutar: "mvn clean install"

Ejecutar: "mvn package"

Ejecutar: "mvn spring-boot:run"

Abrir un navegador y escribir: <http://localhost:8178/tiempo/> (El puerto está cambiado en el archivo "application.properties" de la aplicación, se puede escoger otro).

Ejecución en Tomcat

Para la instalación de la aplicación en Tomcat solo hará falta un servidor Tomcat (preferentemente la versión 8 o superior) y el .war de la aplicación facilitado con este documento o el generado al ejecutar "mvn package" del apartado anterior.

Primeramente arrancar el tomcat usando los archivos startup.sh (distribuciones unix) o startup.bat (windows) que se encuentran en la carpeta bin del programa.

Despues copiar el .war en la carpeta webapps de Tomcat automaticamente la aplicación se desplegará. Si no se modifica el puerto arrancar la aplicación en un navegador con la dirección <http://localhost:8080/tiempo/>

IMAGENES DE LA APLICACIÓN.

Página principal

Consulta el tiempo

[Inicio](#) [Idioma](#) ▼

Aplicación para consultar el tiempo

Introduce el nombre de una ciudad y pulsa sobre consultar para obtener su tiempo.

[Consultar](#)

Busquedas recientes:

[Madrid](#), [Washington](#), [Londres](#), [Cairo](#)

© 2018 Emilio Dominguez.

Página de información del tiempo en el Cairo (Egipto).

El tiempo en Cairo-Egypt

Temperatura :



Humedad: 30

Velocidad del viento: 5 km/h

Mapa de Cairo con sus estaciones metereológicas



Vuelva a buscar una ciudad

Consultar

Busquedas recientes:

Madrid, Washington, Londres, Cairo

INTEGRACIÓN CONTINUA.

INTRODUCCIÓN

Con motivo de la entrevista presencial realizada en la empresa en la cual se habló de que a veces es necesario que los desarrolladores implementen un Jenkins en un servidor, para la integración continua, por ello se decide implementar esta aplicación en un servidor Cloud además de realizar la integración de la aplicación en este Jenkins.

¿PORQUE LA INTEGRACIÓN CONTINUA?

La integración continua es una práctica habitual en el mundo del desarrollo actual, usada principalmente por equipos de desarrollo que han adoptado metodologías ágiles de trabajo. Básicamente, consiste en un proceso por el cual cualquier pequeña mejora en el software se integra rápidamente con el software que se llevará producción

SOFTWARE

Jenkins. Es un software que se usa como un sistema de automatización, capaz de realizar cientos de tareas necesarias para asegurar la calidad del software y facilitar su despliegue o construcción, alguna de las tareas más habituales que se realizan con Jenkins son testar el software, revisar las métricas de calidad del software, despliegue y compilación del software.

En Jenkins usamos los plugins de Maven, Javadoc y JUnit para generar diferente material que será muy útil para ver el estado del proyecto y su evolución.

Git y GitHub. Para la integración continua se usa Git que es un software de control de versiones gratuito. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos. A su vez GitHub es una plataforma de desarrollo colaborativo que permite alojar proyectos usando el control de versiones Git. Típicamente los proyectos que se alojan en GitHub se alojan de forma pública (como en este caso) pero también tiene versiones de pago para alojar proyectos de forma privada. Se usa este servicio por ser gratuito y por su integración con Jenkins.

FUNCIONAMIENTO

Se configura un proyecto en Jenkins llamado "Consulta del tiempo" para que cada vez que se detecte un cambio de código fuente (se realice un comit) en el repositorio del gitHub automaticamente se realizan las siguientes tareas:

- clean install. Que limpia, instala el código fuente del repositorio y ejecuta los tests
- tomcat7:redploy. Compila y carga la aplicación en el tomcat del servidor.
- javadoc:javadoc. Genera la documentación asociada a las clases.

Además se publicarán el javadoc y los tests Junit generados en el jenkins.

ACCESO A APLICACIÓN EN SERVIDOR, A JENKINS Y DIRECCIÓN GITHUB

A través del enlace :

<http://81.46.227.37:8080/tiempo/>

Se podrá acceder a la aplicación (desplegada en un servidor tomcat).

En la dirección :

<http://81.46.227.37:8080/jenkins/>

se puede acceder al jenkins. Se crea un usuario y una contraseña para dar acceso a este servidor y al proyecto para que se puede comprobar su funcionamiento.

Usuario:

evelb

Password:

evelb

El código fuente se encuentra alojado en la dirección:

<https://github.com/milis89uoc/Emilio>

Se puede hacer cualquier cambio en dicho repositorio y se comprobará que automaticamente se despliega en la aplicación gracias a jenkins. El jenkins está configurado para que compruebe si existen cambios en el repositorio cada 10 minutos.