

# Estructura del curso



- **Módulo 1:** Desarrollar aplicaciones web con **Spring MVC** y **Thymeleaf**.
- **Módulo 2:** Persistencia de Datos con **Spring Data JPA**.
- **Módulo 3:** Integración de **Spring MVC** y **Spring Data JPA** en una aplicación web.
- **Módulo 4:** **Spring Security** (seguridad basada en usuarios y roles).
- **Módulo 5:** Desarrollar **Restful Web Services** con **Spring Boot**.

# Proyecto Final – Aplicación para publicar ofertas de trabajo

FileEditViewHistoryBookmarksToolsHelp

EmpleosApp | Creación de Ofertas

itinajero.net:8999/vacantes/edit/6

67%

Bienvenido admin

Salir

Datos de la oferta de trabajo

Nombre

Analista de Crédito y Cobranza

Descripción

Importante Empresa solicita Analista de crédito para actualizar y depurar las cuentas: integración documental

Categoría

Contabilidad/Finanzas

Estatus

Aprobada

Fecha de Publicación

10-06-2019

Salario Ofrecido (Aprox)

21000.0

Imagen

Browse... No file selected.

Destacado

☒ Destacado

☐ No Destacado

Detalles

FileEditViewFormatTableTools

Analista de Crédito y Cobranza

Requisitos:

1. Titulado. Licenciatura en Contabilidad, Administración o afín.

2. 3 años en crédito y cobranza, manejo de Excel a nivel avanzado, SAP (FBL5N)

3. Deseable experiencia en sector farmacéutico


Actividades:

• Aplicación de pagos, compensación de notas de crédito.

• Análisis y actualización en FBL 5N de cada cuenta

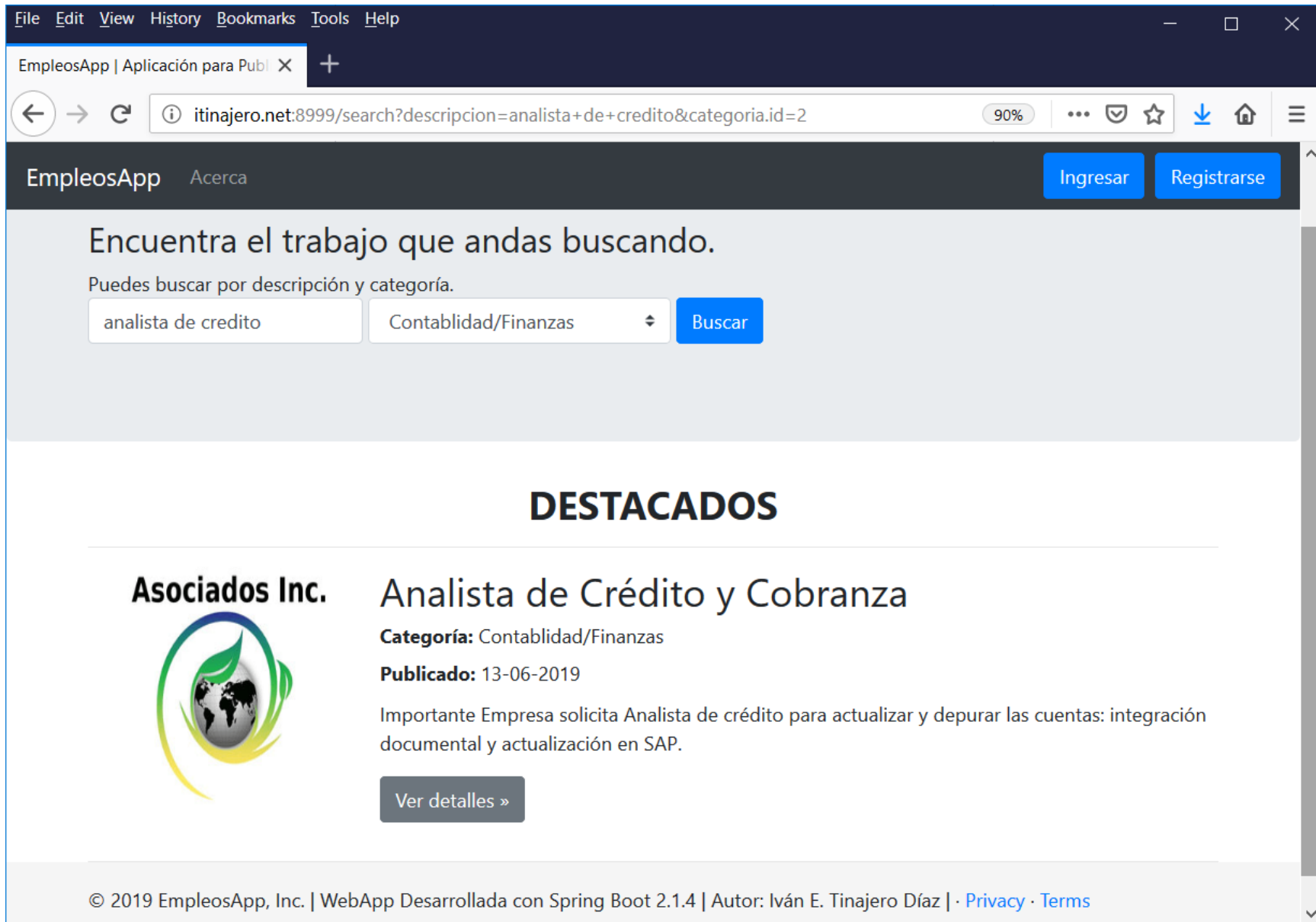
Imagen/Logo

Asociados Inc.

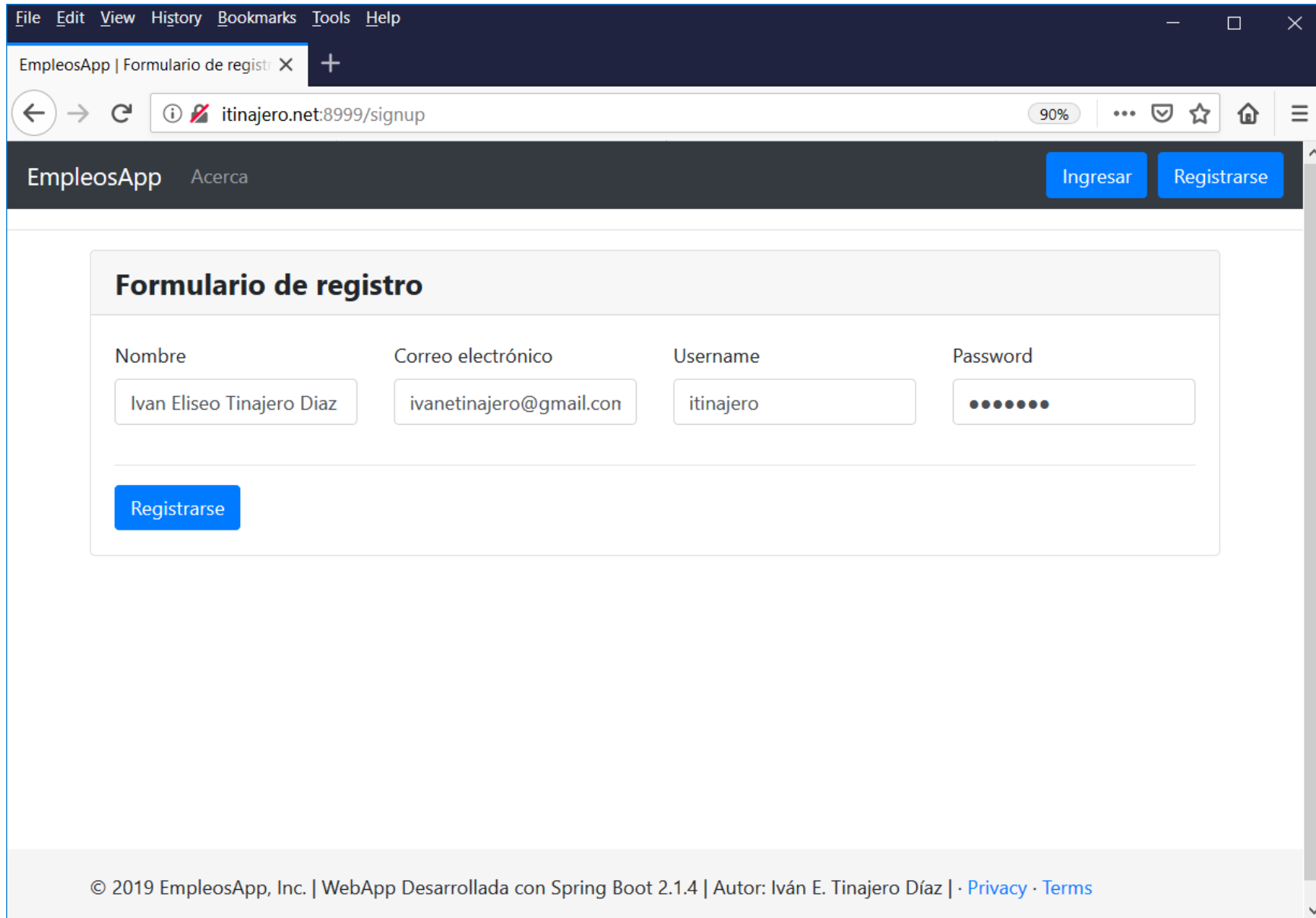


Guardar

# Proyecto Final – Aplicación para publicar ofertas de trabajo



# Proyecto Final – Aplicación para publicar ofertas de trabajo



The image shows a web browser window displaying the registration form for 'EmpleosApp'. The browser's address bar shows the URL 'itinajero.net:8999/signup'. The page has a dark header with the site name 'EmpleosApp' and links to 'Ingresar' and 'Registrarse'. The main content area is titled 'Formulario de registro' and contains four input fields: 'Nombre' (Ivan Eliseo Tinajero Diaz), 'Correo electrónico' (ivanetinajero@gmail.com), 'Username' (itinajero), and 'Password' (masked with dots). A blue 'Registrarse' button is located below the fields. The footer contains copyright information and links to 'Privacy' and 'Terms'.

File Edit View History Bookmarks Tools Help

EmpleosApp | Formulario de registro X +

itinajero.net:8999/signup 90%

EmpleosApp Acerca Ingresar Registrarse

## Formulario de registro

Nombre	Correo electrónico	Username	Password
<input type="text" value="Ivan Eliseo Tinajero Diaz"/>	<input type="text" value="ivanetinajero@gmail.com"/>	<input type="text" value="itinajero"/>	<input type="password" value="....."/>

[Registrarse](#)

© 2019 EmpleosApp, Inc. | WebApp Desarrollada con Spring Boot 2.1.4 | Autor: Iván E. Tinajero Díaz | [Privacy](#) · [Terms](#)

# Proyecto Final – Aplicación para publicar ofertas de trabajo

FileEditViewHistoryBookmarksToolsHelp

EmpleosApp | Aplicar para una oferta X +


←→↻

itinajero.net:8999/solicitudes/create/6

90%⋮🛡️☆🏠☰

EmpleosAppAcercaBienvenido itinajeroSalir

Asociados Inc.



Aplicar para la vacante Analista de Crédito y Cobranza

**Categoría:** Contabilidad/Finanzas

**Publicado:** 2019-06-10 00:00:00.0

**Salario Ofrecido:** 21000.0

Importante Empresa solicita Analista de crédito para actualizar y depurar las cuentas: integración documental y actualización en SAP.

Comentarios

Buenas tardes. Envío mi Curriculum Vitae para aplicar para esta Vacante

Enviar

Curriculum Vitae

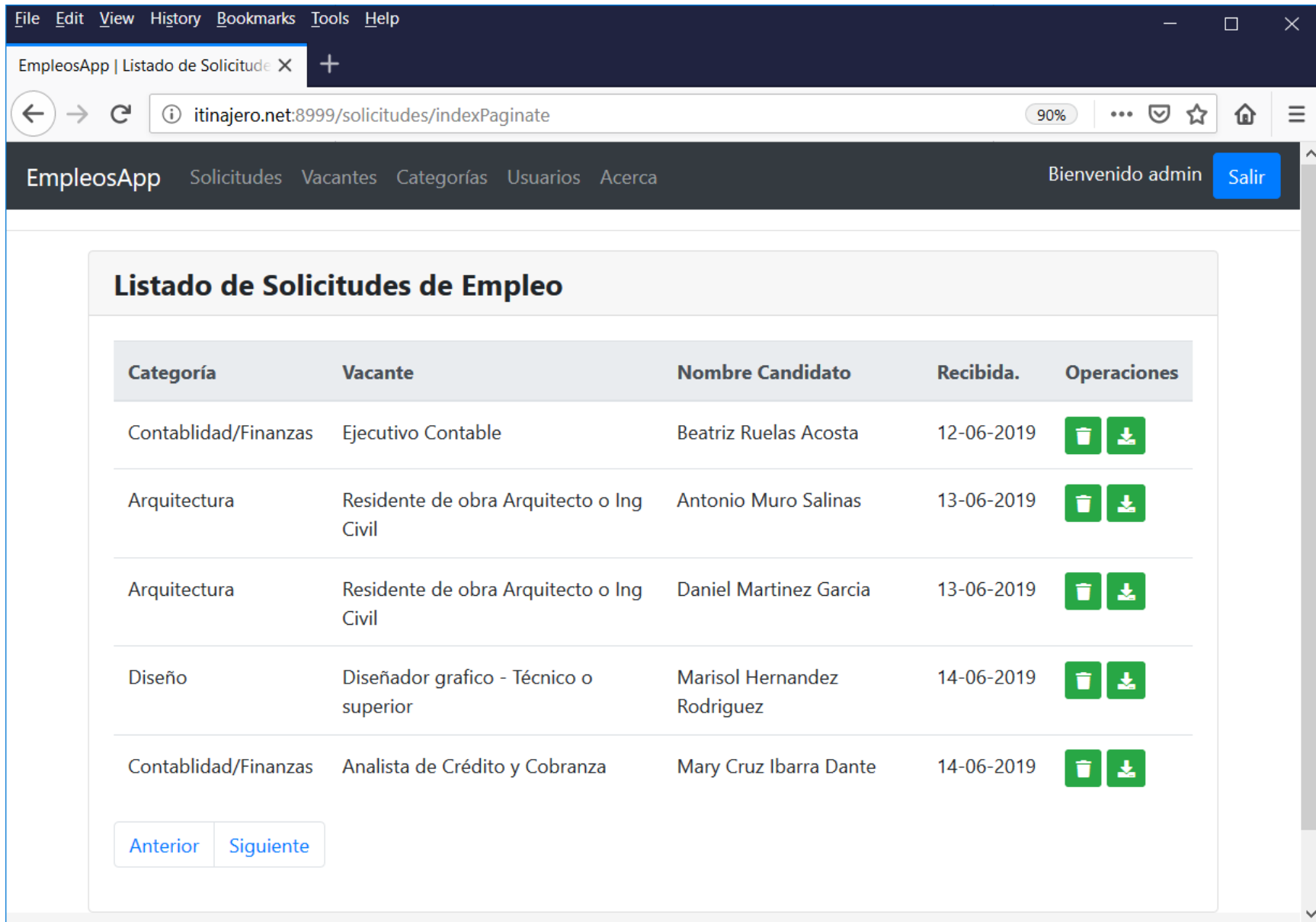
Browse...

curriculum\_vitae.pdf











Solo se permiten archivos [ pdf,doc,docx ]

© 2019 EmpleosApp, Inc. | WebApp Desarrollada con Spring Boot 2.1.4 | Autor: Iván E. Tinajero Díaz | [Privacy](#) · [Terms](#)

# Proyecto Final – Aplicación para publicar ofertas de trabajo



The screenshot displays a web browser window with the address bar showing `itinajero.net:8999/solicitudes/indexPaginate`. The application header includes a navigation menu with links for **EmpleosApp**, **Solicitudes**, **Vacantes**, **Categorías**, **Usuarios**, and **Acerca**. On the right, it says "Bienvenido admin" with a **Salir** button. The main content area is titled "Listado de Solicitudes de Empleo" and contains a table with the following data:

Categoría	Vacante	Nombre Candidato	Recibida.	Operaciones
Contabilidad/Finanzas	Ejecutivo Contable	Beatriz Ruelas Acosta	12-06-2019	 
Arquitectura	Residente de obra Arquitecto o Ing Civil	Antonio Muro Salinas	13-06-2019	 
Arquitectura	Residente de obra Arquitecto o Ing Civil	Daniel Martinez Garcia	13-06-2019	 
Diseño	Diseñador grafico - Técnico o superior	Marisol Hernandez Rodriguez	14-06-2019	 
Contabilidad/Finanzas	Analista de Crédito y Cobranza	Mary Cruz Ibarra Dante	14-06-2019	 

At the bottom of the table, there are two buttons: **Anterior** and **Siguiente**.

DEMO ONLINE → <http://itinajero.net/empleos>

# Desarrollo web con Spring MVC



# ¿Qué es Spring MVC?

- Spring MVC utiliza una arquitectura de aplicaciones siguiendo el patrón de diseño MVC (**M**odel **V**iew **C**ontroller).
- Spring MVC es un framework web basado en Servlets que viene incluido en Spring Framework (spring-webmvc).
- Spring MVC está diseñado siguiendo el patrón de diseño de **Front Controller**.
- En Spring MVC el Front Controller es mejor conocido como **DispatcherServlet**.

## Funciones

- ✓ Enviar las peticiones (requests) a los manejadores (handlers) para que sean procesadas.
  - El default handler son los controladores (@Controller, @RequestMapping).
- ✓ Encargado de resolver las vistas (views).
- Apartir de Spring 3.0 se pueden crear RESTFul Web Services utilizando la anotación **@RestController** y **@PathVariable**.
- Basado en Spring IOC container (Inyección de Dependencias).
- Spring MVC se integra muy fácil con otros proyectos de Spring:
  - ✓ Nosotros integraremos **Spring Boot**, **Spring Data JPA**, **Spring Security**, **Spring REST**, etc.



# ¿Qué es un Controlador en Spring MVC?

- Un Controlador (Controller) en Spring MVC es una clase normal a la cual se le agrega la anotación **@Controller** a nivel de la clase.
- En una aplicación web estos métodos principalmente están marcados con las anotaciones **@GetMapping**, **@PostMapping**, y **@RequestMapping** (Action Controller).
- Los métodos pueden tener cualquier nombre y deben regresar un **String** (nombre de la vista).
- Los métodos son ejecutados al ser invocados por medio de la URL especificada como parámetro en las anotaciones **@GetMapping**, **@PostMapping**, etc.

```
@Controller
public class HomeController {

    @GetMapping("/miUrl")
    public String mostrarHome() {
        // Mi lógica de negocio
        return "home";
    }
}
```

localhost:8080/miUrl

**PETICIÓN TIPO GET**

Quando se utiliza el motor de plantillas **Thymeleaf** se buscará un archivo (vista) llamado **home.html** en el directorio:  
**src/main/resources/templates**

# ¿Qué es Thymeleaf? (1)

- Thymeleaf es un motor de plantillas para aplicaciones web desarrolladas con Java. Es algo similar a los JSPs, con algunas diferencias.
  - ✓ Página Oficial: [www.thymeleaf.org](http://www.thymeleaf.org)
- Comúnmente utilizado con Spring Boot para generar vistas con código HTML para aplicaciones web.
- En un proyecto Spring boot, ya viene configurado Thymeleaf con valores por defecto al momento de agregar la dependencia.

## Configuración

- Agregar la siguiente dependencia al archivo **pom.xml**:

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

# ¿Qué es Thymeleaf? (2)

## Configuración

- Si creamos nuestro proyecto con **Spring Initializr**:

Dependencies

ADD DEPENDENCIES... CTRL + B

Thymeleaf **TEMPLATE ENGINES**

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

- Para utilizar **Thymeleaf** en un archivo HTML se debe agregar el namespace.

**home.html**

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Titulo</title>
</head>
<body>
  <h1 th:text="${mensaje}"></h1>
</body>
</html>
```

**HomeController.java**

```
@Controller
public class HomeController {

    @GetMapping("/")
    public String mostrarHome(Model model) {
        model.addAttribute("mensaje", "Hola Mundo");
        return "home";
    }

}
```

# Iteraciones en Thymeleaf

- En Thymeleaf las iteraciones se pueden realizar con la expresión **th:each**
  - ✓ Similar a un for en Java.
- Esta expresión puede iterar sobre diferentes tipos de datos como:
  - ✓ List
  - ✓ Map
  - ✓ Iterable
  - ✓ Map

Ejemplo:

Vista(detalle.html)

```
<tr th:each="tmpEmp: ${empleos}">

    <td th:text="${tmpEmp}" />

</tr>
```

Controlador

```
@GetMapping("/detalle")
public String mostrarDetalle(Model model) {
    List<String> lista = new LinkedList<>();
    lista.add("Ingeniero de Sistemas");
    lista.add("Auxiliar de Contabilidad");
    model.addAttribute("empleos", lista);
    return "detalle";
}
```

# Condicionales en Thymeleaf

## ➤ Operador Elvis (?:)

El operador Elvis permite renderizar texto DENTRO de un elemento HTML, dependiendo de una expresión Booleana. Es muy similar al operador ternario en otros lenguajes de programación.

Ejemplo:

```
<td th:text="${usuario.estatus == 1} ? 'ACTIVO' : 'BLOQUEADO'" />
```

Valor renderizado, si la  
expresión es verdadera.

Valor renderizado, si la  
expresión es falsa.

## ➤ if – unless

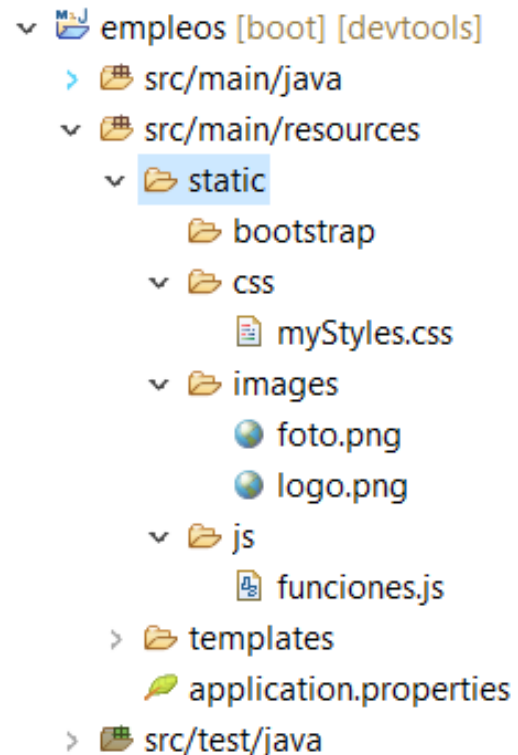
La expresión **if - unless** permite renderizar un elemento HTML, dependiendo de una expresión Booleana. Es muy similar a un **if-else** en otros lenguajes de programación.

Ejemplo:

```
<td>
  <span th:if="${alumno.genero == 'F'}"> Femenino </span>
  <span th:unless="${alumno.genero == 'F'}"> Masculino </span>
</td>
```

# Urls relativas al ContextPath en Thymeleaf

- Las URLs relativas al ContextPath son las que son relativas al directorio raíz (ROOT) de una aplicación web, una vez que están publicadas en el servidor.
- Las URLs relativas al ContextPath deben iniciar con "/" cuando vayamos a formar una URL para referenciar un recurso (imagenes, CSS, JS, PDF, etc) en nuestra aplicación.
- En un proyecto web cuando se utiliza **Thymeleaf** como motor de plantillas, los recursos estáticos deben guardarse en el directorio **src/main/resources/static**



## Ejemplos:

- Para incluir el archivo CSS myStyles.css en una vista se utilizaría la siguiente expresión:  
`<link th:href="@{/css/myStyles.css}" rel="stylesheet">`
- Para incluir el archivo Javascript funciones.js en una vista se utilizaría la siguiente expresión:  
`<script th:src="@{/js/funciones.js}"></script>`
- Para incluir la imagen foto.png en una vista, se utilizaría la siguiente expresión:  
``

Para incluir archivos Javascript y CSS vía CDN (*Content Delivery Network*) se utiliza la sintaxis estándar (sin expresiones Thymeleaf).

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
```