

Integración de Spring MVC y Spring Data JPA



Configurar directorios de recursos estáticos adicionales

- Por defecto en un proyecto de **Spring Boot** con **Thymeleaf** los recursos estáticos (css, javascript, images) están configurados en el directorio:
 - ✓ **src/main/resources/static**
- ¿Qué pasa si necesitamos mapear recursos estáticos de otros directorios adicionales?

```
@Configuration
public class WebConfig implements WebMvcConfigurer{
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        //registry.addResourceHandler("/logos/**").addResourceLocations("file:/empleos/img-vacantes/"); // Linux
        registry.addResourceHandler("/logos/**").addResourceLocations("file:c:/empleos/img-vacantes/"); // Windows
    }
}
```

Mapping URL

Si tenemos por ejemplo la imagen “logo.png” en el directorio c:/empleos/img-vacantes/ podríamos llamar la imagen con la siguiente URL:

<http://localhost:8080/logos/logo.png>

Obligatorio el prefijo **file:**

Integración de paginación

El método `findAll`, está sobrecargado con un parámetro de tipo `Pageable`. Ejemplo:

Clase de servicio

```
@Override
public Page<Vacante> buscarTodas(Pageable page) {
    return vacantesRepo.findAll(page);
}
```

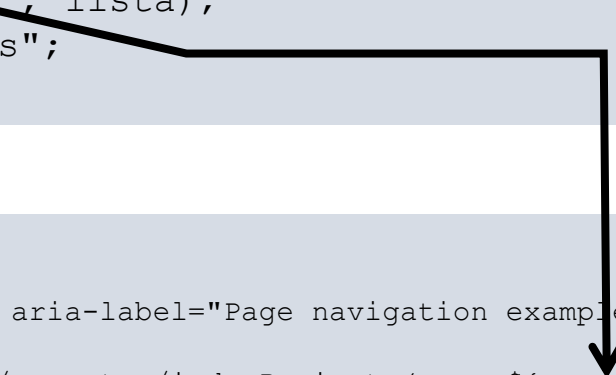
Controller

```
@GetMapping(value = "/indexPaginate")
public String mostrarIndexPaginado(Model model, Pageable page) {
    Page<Vacante> lista = serviceVacantes.buscarTodas(page);
    model.addAttribute("vacantes", lista);
    return "vacantes/listVacantes";
}
```

Vista (links de paginación)

```
<table> . . . </table>
















<nav th:if="${vacantes.totalPages > 0}" aria-label="Page navigation example" >
  <ul class="pagination">
    <li class="page-item"><a th:href="@{/vacantes/indexPaginate(page=${vacantes.number - 1})}" class="page-link">Anterior</a></li>
    <li class="page-item"><a th:href="@{/vacantes/indexPaginate(page=${vacantes.number + 1})}" class="page-link">Siguiente</a></li>
  </ul>
</nav>
```



Propiedad para especificar el tamaño de la página (application.properties)

```
spring.data.web.pageable.default-page-size=5
```

Resultado Final – Listado de Vacantes con paginación

Categoría	Nombre	Fecha Pub.	Estatus	Destacada	Operaciones
Contabilidad/Finanzas	Analista Contable.	04-06-2019	Aprobada	NO	 
Computacion/TI	Administrador de datos en la nube	04-06-2019	Aprobada	NO	 
Mercadotecnia	Coordinador de Marketing	05-06-2019	Aprobada	NO	 
Recursos Humanos	Gerente de Recursos Humanos	05-06-2019	Aprobada	NO	 
Logistica/transportes	Jefe de Almacén	06-06-2019	Aprobada	NO	 
Contabilidad/Finanzas	Analista de Crédito y Cobranza	06-06-2019	Aprobada	SI	 
Diseño	Diseñador grafico - Técnico o superior	06-06-2019	Aprobada	SI	 
Recursos Humanos	Gerente de Recursos Humanos	07-06-2019	Aprobada	SI	 
Arquitectura	Residente de obra Arquitecto o Ing Civil	07-06-2019	Aprobada	SI	 
Computacion/TI	Administrador de redes y Sistema Operativo Red Hat	08-06-2019	Aprobada	SI	 

[Anterior](#)[Siguiete](#)

NOTA: No olvides actualizar el link del menú (**menu.html**) para acceder al listado de vacantes. Ahora en el link debería ser:

```
<a class="nav-link" th:href="@{/vacantes/indexPaginate}">Vacantes</a>
```