



**Universidad Nacional Autónoma de
México**

**Dirección General de Cómputo de
Tecnologías de Información y
Comunicación**

Diplomado

Desarrollo de Sistemas con Tecnología Java

Ejercicio Cinco

“i18n, Imágenes, Errores, Pdf y Correos”

Mtro. ISC. Miguel Ángel Sánchez Hernández

Tabla de contenido

1. Abrir proyecto con IntelliJ.....	3
2. Agregar mensajes en otros idiomas.....	3
3. ConfigMVC	3
4. Cambiar mensajes en el controlador.....	5
5. Subir imágenes.....	6
6. Redirección general de errores.....	8
7. Redirección particular de un error	9
8. Creación de pdfs.....	10
9. Mandar correo electrónico	15

1. Abrir proyecto con IntelliJ

Para abrir un proyecto, hacemos lo siguiente:

1. Señalamos File→Open
2. Buscamos la ruta donde se encuentra el proyecto spring-web
3. Presionamos Ok
4. Actualizar maven de ser necesario

2. Agregar mensajes en otros idiomas

El proceso es el siguiente:

1. Señalamos a resources.
2. Click derecho→New→Resource Bundle.
3. Resource bundle base name: messages.
4. Click en el boton de “+”.
5. En la ventana emergente de Add locales, agregamos “en”, sin las dobles comillas.
6. Ok

Los archivos que se agregan deben de tener la misma llave que los otros pero su valor al idioma que necesitamos cambiar. Por ejemplo messages_en.properties

```
Length.comprador.nombre= Size of {0} must be in the range of {2} and {1}
Pattern.comprador.correo= The email must end with dgctic
NotEmpty.comprador.apellidoPaterno=Paternal Last Name must not be empty
NoEspacioNoVacio.comprador.apellidoMaterno=Maternal Surname must not be empty
Duplicado.comprador.correo=Duplicate email
```

3. ConfigMVC

Crear la clase ConfigMVC.java en la carpeta util.

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.LocaleResolver;
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;
import org.springframework.web.servlet.i18n.SessionLocaleResolver;

@Configuration
public class ConfigMVC implements WebMvcConfigurer {
    @Bean
    public LocaleResolver localeResolver(){
        SessionLocaleResolver localeResolver=new SessionLocaleResolver();
        localeResolver.setDefaultLocale(new Locale("es","MX"));
        return localeResolver;
    }
}
```

```

@Bean
public LocaleChangeInterceptor localeChangeInterceptor(){
    LocaleChangeInterceptor localeChangeInterceptor=new LocaleChangeInterceptor();
    localeChangeInterceptor.setParamName("lang");
    return localeChangeInterceptor;
}
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(localeChangeInterceptor());
}
}

```

plantilla.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
    lang="es">
<head th:fragment="head">
    <meta charset="UTF-8">
    <title>Spring</title>

    <link href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" rel="stylesheet">
    <link th:href="@{/bootstrap/css/bootstrap.min.css}" rel="stylesheet">
    <link th:href="@{/tema/miestilo.css}" rel="stylesheet">
</head>
<body>
<header th:fragment="header">
    <div class="container-fluid text-center">
        <div class="row">
            <div class="col-lg-2 navbar bg-dark">
                <a class="navbar navbar-brand" style="color:white" th:href="@{/}">Spring Web MVC</a>
                <a class="btn btn-outline-primary" th:href="@{/ (lang=es_MX)}">es</a>
                <a class="btn btn-outline-primary" th:href="@{/ (lang=en_US)}">en</a>
            </div>
            .
            .
            .
        </div>
    </div>

```

4. Cambiar mensajes en el controlador

Crear una clase en el paquete util que se llame AppConfig

```
package dgctic.core.util;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.support.ResourceBundleMessageSource;

@Configuration
public class AppConfig {
    @Bean
    public ResourceBundleMessageSource messageSource() {
        var source = new ResourceBundleMessageSource();
        source.setBasenames("messages");
        source.setUseCodeAsDefaultMessage(true);
        return source;
    }
}
```

CompradorController.java

```
@Controller
@RequestMapping(value = "comprador")
public class CompradorController {
    @Autowired
    ResourceBundleMessageSource mensaje;

    .
    .
    .
    @PostMapping("salvar-comprador")
    public String salvarObjectThymeleaf(@Valid @ModelAttribute("comprador") CompradorEntity comprador, BindingResult result,
        Model model) {
        if (result.hasErrors()) {
            model.addAttribute("contenido", "Error en la entrada de datos");
            return "comprador/alta-comprador";
        }
        try {
            compradorService.guardar(comprador);
        } catch (DataIntegrityViolationException exception) {
            ObjectError er = new ObjectError("correo", mensaje.getMessage("Duplicado.comprador.correo",
                null, Locale.getDefault()));
            result.addError(er);
            model.addAttribute("success", "Ya está registrado la tarjeta o correo o teléfono");
            return "comprador/alta-comprador";
        }
        return "comprador/alta-comprador";
    }
}
```

5. Subir imágenes

Lo primer es crear un directorio en la ruta de tu preferencia, por ejemplo "C:\imagenes", abrimos ConfigMVC, agregamos el siguiente código hasta abajo.

```
@Configuration
public class ConfigMVC implements WebMvcConfigurer {
    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        WebMvcConfigurer.super.addResourceHandlers(registry);
        registry.addResourceHandler("/imagenes/**")
            .addResourceLocations("file:/C:/imagenes/");
    }
}
```

Crear una clase que se llame Archivos.java en el paquete util.

```
package dgtic.core.util;
import org.springframework.web.multipart.MultipartFile;
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class Archivos {
    public static String almacenar(MultipartFile multipartFile,String direccion){
        String nombre=multipartFile.getOriginalFilename();
        Path path= Paths.get(direccion);
        direccion=path.toFile().getAbsolutePath();

        File imagen=new File(direccion+"/"+nombre);
        Path finalRuta=Paths.get(imagen.getAbsolutePath());
        try {
            byte[] bytes=multipartFile.getBytes();
            Files.write(finalRuta,bytes);

            //multipartFile.transferTo(imagen);
            return nombre;
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Error al subir archivo");
            return null;
        }
    }
    public static void renombrar(String direccion,String viejo,String nuevo){
        File archivoViejo=new File(direccion+"/"+viejo);
        File archivoNuevo=new File(direccion+"/"+nuevo);
        archivoViejo.renameTo(archivoNuevo);
    }
}
```

alta-especie.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
  lang="es">
<head th:replace="~{plantillas/plantilla :: head}"></head>
<body>
  <header th:replace="~{plantillas/plantilla :: header}"></header>
  <main class="container text-center shadow-lg p-3 mb-5 bg-body-tertiary rounded">
    <h1 th:text="{contenido}"></h1>
    <div class="col-md-9">
      <div class="form-floating">
        <form th:action="@{/especie/salvar-imagen}" method="post" enctype="multipart/form-data">
          <div class="form-group">
            <label for="imagenarchivo">Subir imagen (opcional)</label>
            <div class="input-group mb-1">
              <input type="file" class="form-control" id="imagenarchivo" name="imagenarchivo"/>
              <button type="submit" class="btn btn-primary">Subir imagen</button>
            </div>
          </div>
        </form>
      </div>
      <div>
        <form th:action="@{/especie/salvar-especie}" th:object="{especie}" method="post">
          <div th:if="{#fields.hasErrors('*)}" class="alert alert-danger" role="alert">
            Verifique los siguientes errores
            <ul>
              <li th:each="err : {#fields.errors('*)}" th:text="{err}"></li>
            </ul>
          </div>
          <div class="row">
            <div class="col-md-4">
              <div class="form-group">
                <label for="nombre">Nombre de la especie</label>
                <input type="text" class="form-control" id="nombre" name="nombre"
                  placeholder="Nombre de la especie" required="required"
                  th:field="*{nombre}">
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label for="lote">Lotes Disponibles</label>
                <select class="form-control" th:field="*{lote}" id="lote" name="lote" required="required">
                  <option th:each="sel : {selectLote}" th:value="{sel.id_lte}" th:text="Lote: '{sel.id_lte}' + ' Fecha: '{sel.fecha} + ' Precio: '{sel.precioKiloSalida}'>
                </select>
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label for="tipo">Tipos</label>
                <select class="form-control" th:field="*{tipo}" id="tipo" name="tipo" required="required">
                  <option th:each="sel : {selectTipo}" th:value="{sel.id_tpo}" th:text="{sel.nombre}">
                </select>
              </div>
            </div>
            <div class="col-md-3">
              <br/>
              <button type="submit" class="btn btn-primary">Guardar</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </main>

```

```

</main>
<footer th:replace=~{plantillas/plantilla :: footer}"></footer>
</body>
</html>

```

EspecieController.java

```

@Controller
@RequestMapping(value="especie")
@SessionAttributes("especie")
public class EspecieController {
    @Value("${ejemplo.imagen.ruta}")
    private String archivoRuta;
    .
    .
    .
    PostMapping("salvar-especie")
    public String salvarObjectThymeleaf(@Valid @ModelAttribute("especie") EspecieEntity especie, BindingResult result,
        Model model, RedirectAttributes flash,
        SessionStatus session) {
        if (result.hasErrors()) {
            model.addAttribute("contenido", "Error en la entra de datos");
            return "especie/alta-especie";
        }
        especieService.guardar(especie);
        session.setComplete();
        flash.addFlashAttribute("success", "Lote se almaceno con éxito");
        return "redirect:/especie/lista-especie";
    }
    @PostMapping(value = "salvar-imagen")
    public String guardar(@RequestParam("imagenarchivo") MultipartFile multipartFile,
        HttpSession session, Model model){
        if(!multipartFile.isEmpty()){
            String imagenNombre= Archivos.almacenar(multipartFile,archivoRuta);
            if(imagenNombre!=null){
                EspecieEntity especie=(EspecieEntity) session.getAttribute("especie");
                especie.setImagen(imagenNombre);
            }
        }
        List<LoteEntity> lote=loteService.buscarLotes();
        List<TipoEntity> tipo=tipoService.buscarTipo();
        model.addAttribute("selectLote",lote);
        model.addAttribute("selectTipo",tipo);
        return "especie/alta-especie";
    }
}

```

6. Redirección general de errores

Modificar el archivo application.properties y agregar al final la siguiente línea.

```

#cambio de la página de error por defecto
server.error.whitelabel.enabled=false

```


Crear el archivo en el directorio paginas llamado error.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      lang="es">
<head th:replace="~{plantillas/plantilla :: head}"></head>
<body>
<div class="container text-center">
  <h1 class="text-secondary border border-success border-top-0">Recurso no encontrado</h1>
  <br/>
  <div class="container py-4 text-center">
    <div class="card by-danger">
      <div class="card-body">
        <h4 class="card-title">
          El recurso que solicitas no esta disponible ó ocurrió un error interno
        </h4>
        <a class="btn btn-outline-primary" th:href="@{/}">Inicio</a> </li>
      </div>
    </div>
  </div>
</div>

<footer th:replace="~{plantillas/plantilla :: footer}"></footer>
</body>
</html>
```

7. Redirección particular de un error

Modificar CompradorController.java cambiando el método salvarObjectThymeleaf por el código siguiente:

```
@PostMapping("salvar-comprador")
public String salvarObjectThymeleaf(@Valid @ModelAttribute("comprador") CompradorEntity comprador, BindingResult result,
                                     Model model) {
    if (result.hasErrors()) {
        model.addAttribute("contenido", "Error en la entra de datos");
        return "comprador/alta-comprador";
    }
    compradorService.guardar(comprador);
    return "comprador/alta-comprador";
}

@ExceptionHandler(DataIntegrityViolationException.class)
public String errorRuntime(DataIntegrityViolationException e,
                           Model model){
    String msg= mensaje.getMessage("Duplicado.comprador.correo",
        null,Locale.getDefault());
    model.addAttribute("explicacion",msg);
    return "error-general";
}
```

Crear el archivo error-general.html en el directorio paginas.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      lang="es">
<head th:replace=~{plantillas/plantilla :: head}"></head>
<body>
<div class="container text-center">
    <h1 class="text-secondary border border-success border-top-0">Problema no esperado</h1>
    <br/>
    <div class="container py-4 text-center">
        <div class="card by-danger">
            <div class="card-body">
                <h4 class="card-title">
                    <label th:text="\${explicacion}""></label>
                </h4>
                <a class="btn btn-outline-primary" th:href="@{/}">Inicio</a> </li>
            </div>
        </div>
    </div>
</div>
</div>
<div>
<div>
<div>
```

8. Creación de pdfs

Agregar la dependencia en pow.xml de openpdf

```
<dependency>
  <groupId>com.github.librepdf</groupId>
  <artifactId>openpdf</artifactId>
  <version>1.3.30</version>
</dependency>
```

Agregar la siguiente línea en `application.properties`

```
#pdf
spring.mvc.contentnegotiation.favor-parameter=true
spring.mvc.contentnegotiation.media-types.pdf=application/pdf
```

Modificar EspecieService.java

```
public interface EspecieService {
    List<EspecieEntity> buscarEspecie(Integer id);
    Page<EspecieEntity> buscarEspecie(Pageable pageable);
    .
}
```

Modificar EspecieServiceImpl

```
@Service
public class EspecieServiceImpl implements EspecieService{
    @Autowired
    EspecieRepository especieRepository;

    @Override
    public List<EspecieEntity> buscarEspecie(Integer id) {
        return especieRepository.findAllById(Collections.singleton(id));
    }

    @Override
    public Page<EspecieEntity> buscarEspecie(Pageable pageable) {
        return especieRepository.findAll(pageable);
    }
    .
    .
    .
}
```

Crear la clase EspeciePdf.java en el paquete util

```
package dgtic.core.util;
import com.lowagie.text.Document;
import com.lowagie.text.Image;
import com.lowagie.text.Phrase;
import com.lowagie.text.pdf.PdfPCell;
import com.lowagie.text.pdf.PdfPTable;
import com.lowagie.text.pdf.PdfWriter;
import dgtic.core.model.entities.EspecieEntity;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.stereotype.Component;
import org.springframework.web.servlet.view.document.AbstractPdfView;

import java.awt.*;
import java.io.FileOutputStream;
import java.util.List;
import java.util.Map;

@Component("especie/especie-pdf")
public class EspeciePdf extends AbstractPdfView {
    @Override
    protected void buildPdfDocument(Map<String, Object> model, Document document,
        PdfWriter writer,
        HttpServletRequest request, HttpServletResponse response) throws Exception {

        List<EspecieEntity> datos= (List<EspecieEntity>) model.get("datos");
        String ruta=((String)model.get("ruta"))+datos.get(0).getImagen();
        String rutaAlm=ruta+".pdf";
        PdfWriter.getInstance(document,new FileOutputStream(rutaAlm));
        document.open();
        PdfPTable tabla=new PdfPTable(2);
        PdfPCell celda=new PdfPCell(new Phrase("Id de la Especie"));
        celda.setBackgroundColor(new Color(167,218,255));
        celda.setPadding(8f);
        tabla.addCell(celda);
        tabla.addCell("Nombre");
        for (EspecieEntity r:datos) {
            tabla.addCell(String.valueOf(r.getLote().getId_lte()));
            tabla.addCell(r.getNombre());
        }
        document.add(tabla);
        Image jpg = Image.getInstance(ruta);
        document.add(jpg);
        document.close();
    }
}
```

Agregar las siguientes líneas em EspecieController.java

```

@GetMapping(value = "buscar-especie-nombre/{dato}", produces = "application/json")
public @ResponseBody List<EspecieEntity> buscarCliente(@PathVariable String dato) {
    return especieService.buscarEspeciePatron(dato);
}

@GetMapping(value = "buscar-especie")
public String buscarEspeciePagina(Model model) {
    EspecieEntity entity = new EspecieEntity();
    List<EspecieEntity> lista=new ArrayList<>();
    model.addAttribute("especie", entity);
    model.addAttribute("datos", lista);
    model.addAttribute("contenido", "Buscar especie por nombre");
    return "especie/buscar-especie";
}

@PostMapping("buscar-especie")
public String buscarReservaconTabla(@Valid @ModelAttribute("especie") EspecieEntity especie
    , BindingResult result,Model model, RedirectAttributes flash) {
    List<EspecieEntity> especies=especieService.buscarEspecie(especie.getId_epe());
    model.addAttribute("datos",especies);
    return "especie/buscar-especie";
}

@GetMapping("pdf")
public String generarPdf(Model model, HttpSession session){
    //obteniendo la variable de sesión
    EspecieEntity esp=(EspecieEntity)session.getAttribute("especie");
    List<EspecieEntity> lista=especieService.buscarEspecie(esp.getId_epe());
    model.addAttribute("datos",lista);
    model.addAttribute("ruta",archivoRuta);
    return "especie/especie-pdf";
}

```

Crear el archivo buscar-especie.html en la carpeta especie

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
  lang="es">
<head th:replace=~{plantillas/plantilla :: head}"></head>
<body>
  <header th:replace=~{plantillas/plantilla :: header}"></header>
  <main class="container text-center shadow-lg p-3 mb-5 bg-body-tertiary rounded">
    <h1 th:text="{contenido}"></h1>
    <div class="card">
      <h4 class="card-header"><strong>Nombre de la especie</strong></h4>
      <div class="card-body">
        <form th:action="@{/especie/buscar-especie}" th:object=${especie} method="post">
          <div class="row">
            <div class="col-md-5">
              <div class="form-group">
                <label for="nombre">Nombre de la Especie</label>
                <input type="text" class="form-control" id="nombre" name="nombre"
                  placeholder="Nombre de la especie" required="required"
                  th:field="*{nombre}">
              </div>
            </div>
            <div class="col-md-3">
              <br/>
              <button type="submit" class="btn btn-primary">Buscar</button>
            </div>
          </div>
          <input type="hidden" id="idEspecie" name="idEspecie" th:field="*{id_epe}" />
        </form>
        <a th:if="{datos.size()}>0" class="btn btn-primary btn-xs float-right" th:href="@{/especie/pdf(format=pdf)}"
          target="_blank">Pdf</a>
        <table th:if="{datos.size()}>0" class="table table-striped table-hover table-sm">
          <thead class="table-dark table-bordered">
            <tr>
              <th scope="col">Nombre</th>
              <th scope="col">Imagen</th>
            </tr>
          </thead>
          <tbody>
            <tr th:each="esp: ${datos}">
              <td th:text="{esp.nombre}"></td>
              <td>
                <br/>
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </main>
  <footer th:replace=~{plantillas/plantilla :: footer}"></footer>
  <script th:replace=~{especie/js/autocompletar :: jsespecie}"></script>
</body>
</html>

```

Agregar los siguientes archivos en las carpetas siguientes:

jquery-ui.min.css	→	static/bootstrap/css	
jquery-ui.min.js	→	static/bootstrap/js	
autocompletar	→	paginas/especie/js	“Este ultimo crear la carpeta js dentro de especie”

9. Mandar correo electrónico

Configurar una cuenta de correo electrónico de Gmail para mandar correos en una aplicación, siguiente los siguientes pasos:

- Acceder a la cuenta de Google
- Administrador de cuentas de Google
- Seguridad
- verificación 2 pasos activar
- hasta abajo "contraseñas de aplicaciones"
- Nombre de la aplicación
- Copiar la contraseña

Modificar EspecieController.java

```
@GetMapping("pdf")
public String generarPdf(Model model, HttpSession sesion){
    //obteniendo la variable de sesión
    EspecieEntity esp=(EspecieEntity)sesion.getAttribute("especie");
    List<EspecieEntity> lista=especieService.buscarEspecie(esp.getId_epe());
    model.addAttribute("datos",lista);
    model.addAttribute("ruta",archivoRuta);
    return "especie/especie-pdf";
}

@GetMapping("correo")
private String gmail(Model model, HttpSession sesion ){
    EspecieEntity esp=(EspecieEntity)sesion.getAttribute("especie");
    System.out.println(esp.getImagen()+"<----");
    List<EspecieEntity> lista=especieService.buscarEspecie(esp.getId_epe());
    String gmail="desarrollo.spring.mas@gmail.com";
    String pswd="Tu password";
    Properties p=System.getProperties();
    p.setProperty("mail.smtps.host","smtp.gmail.com");
    p.setProperty("mail.smtps.socketFactory.class","javax.net.ssl.SSLSocketFactory");
    p.setProperty("mail.smtps.socketFactory.fallback","false");
    p.setProperty("mail.smtp.port","465");
    p.setProperty("mail.smtp.socketFactory.port","465");
    p.setProperty("mail.smtps.auth","true");
    p.setProperty("mail.smtp.ssl.trust","smtp.gmail.com");
    p.setProperty("mail.smtps.ssl.trust","smtp.gmail.com");
    p.setProperty("mail.smtp.ssl.quitwait","false");
    //construcción del html
    String cadena="<h2>Nombre de la especie</br>";
    for (EspecieEntity s:lista ) {
```

```
cadena+="<h2>"+
    s.getNombre()+
    "</h2></br>";
}
try{
    Session session=Session.getInstance(p,null);
    MimeMessage message=new MimeMessage(session);

    MimeBodyPart texto=new MimeBodyPart();
    texto.setContent(cadena,"text/html;charset=utf-8");
    //adjuntar la imagen
    BodyPart adjunto=new MimeBodyPart();
    String r=archivoRuta+lista.get(0).getImagen()+".pdf";
    adjunto.setDataHandler(new DataHandler(new FileDataSource(r)));
    adjunto.setFileName("especie.pdf");
    Multipart multiple=new MimeMultipart();
    multiple.addBodyPart(texto);
    multiple.addBodyPart(adjunto);

    message.setRecipients(Message.RecipientType.TO,
        InternetAddress.parse("al correo que quieres mandar",false));
    message.setSubject("Especie Registrada en B.D");
    message.setContent(multiple);
    message.setSentDate(new Date());

    Transport transport=(Transport)session.getTransport("smtps");
    transport.connect("smtp.gmail.com",gmail,pswd);
    transport.sendMessage(message,message.getAllRecipients());
    transport.close();

} catch (Exception e){
    e.printStackTrace();
}
model.addAttribute("success","El correo se mando con éxito");
model.addAttribute("datos",lista);
model.addAttribute("ruta",archivoRuta);
return "especie/buscar-especie";
}
```


Modificar buscar-especie.html

[illegible]