

# GUÍA DE REFERENCIA

ISRAEL DE LAMO BLAS

FELÍ DÍAS DE ALDA

## ÍNDICE

### Tabla de contenido

GUÍA DE REFERENCIA .....	1
¿QUÉ VOY A ENCONTRAR EN ESTE DOCUMENTO? .....	3
MENSAJES DE ERROR .....	4
Mensaje de error LOG del fichero “registroDeApp.log” .....	4
Mensaje de error generado por Alertas.class.....	5
Mensajes de información de Alertas.class .....	6

## ¿QUÉ VOY A ENCONTRAR EN ESTE DOCUMENTO?

Al contrario que en el caso de las guías rápidas, las guías de referencias suelen estar pensadas para **usuarios experimentados** y avanzados.

Por ello, es habitual que contengan información relacionadas con aspectos más técnicos:

- Mensajes de error y como recuperarse de ellos.
- Guía con comandos para usar la aplicación en modo comando.
- Tipos de datos de entrada que son permitidos en la aplicación.

## MENSAJES DE ERROR

El programa puede lanzarnos tres tipos de mensaje de error; tres de tipo log y otro de tipo ventana de alertaFX.

Los mensajes de alertaFX pueden a su vez ser de tres tipos; información, error y error no gestionado.

Vamos a desglosar cada uno de ellos para poder hacer un mejor diagnostico de cada problema.

### Mensaje de error LOG del fichero “registroDeApp.log”

Este fichero se genera a medida que se usa el programa y corre el peligro de ocupar tanto espacio como tenga la unidad. Hay que tener en cuenta que su fin es puramente didáctico.

```
<appender name="FILE_GIGANTE"
class="ch.qos.logback.core.FileAppender">
  <file>registroDeApp.log</file>  <!-- Ruta del archivo de log -->
  <encoder>
    <pattern>%d{yyyy-MM-dd HH:mm:ss} - %msg%n</pattern>
  </encoder>
</appender>
<!-- Configuración del logger -->
<logger name="es.israeldelamo.transportes" level="INFO">
  <appender-ref ref="CONSOLE"/>
  <appender-ref ref="FILE_GIGANTE"/>
  <appender-ref ref="FILE_ROLLER"/>
</logger>
<!-- Configuración del root logger -->
<root level="DEBUG">
  <appender-ref ref="CONSOLE"/>
  <appender-ref ref="FILE_GIGANTE"/>
  <appender-ref ref="FILE_ROLLER"/>
</root>
```

El fichero va guardando cada passo del static logger que se encuentra por todo el programa, dejando un rastro claro del trabajo del usuario.

```
2025-03-23 13:30:48 - Database change: is 'transportes'
2025-03-23 13:30:48 - execute query: set
sql_mode=CONCAT(@@sql_mode,',STRICT_TRANS_TABLES'),session_track_syste
m_variables =
CONCAT(@@global.session_track_system_variables,',tx_isolation'),NAMES
utf8mb4
2025-03-23 13:30:48 - System variable change:
character_set_connection = utf8mb4
2025-03-23 13:30:48 - System variable change:  character_set_client =
utf8mb4
2025-03-23 13:30:48 - System variable change:  character_set_results =
utf8mb4
2025-03-23 13:30:48 - execute query: SELECT cp, localidad,
```

```
provincia,pais FROM CODIGOS_POSTALES WHERE (localidad=?) AND  
(provincia =?) AND (pais=?)
```

En este extracto se ve la primera parte la fecha de trabajo. Seguidamente la hora. Y los mensajes son sobre todo salidas de las consultas de la base de datos.

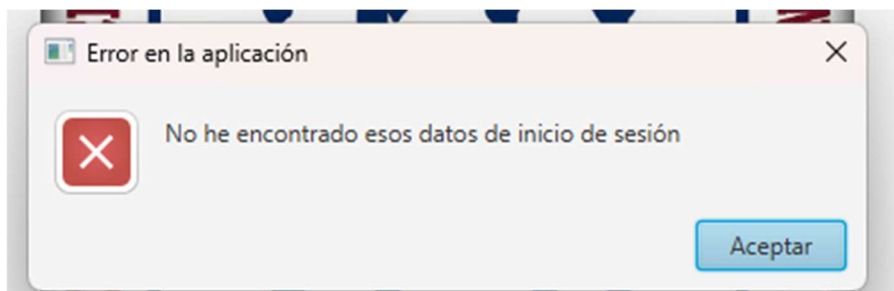
En caso de encontrarse un error, por ejemplo, nullpointer quedaría recogido.

```
2025-03-23 15:53:25,545 [JavaFX Application Thread] ERROR e.i.t.c.GestionAdministradorController - Ni siquiera he podido cr  
java.lang.NullPointerException: Cannot invoke "es.israeldelamo.transportes.modelos.ModeloNaturalezasPeligrosas.getCodigo_na  
    at es.israeldelamo.transportes.controladores.GestionAdministradorController.onBotonNuevoDocumentoCapacitacion(GestionAd  
    at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)  
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)  
    at com.sun.javafx.reflect.Trampoline.invoke(MethodUtil.java:72)  
    at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103)  
    at java.base/java.lang.reflect.Method.invoke(Method.java:580)  
    at com.sun.javafx.reflect.MethodUtil.invoke(MethodUtil.java:270)  
    at com.sun.javafx.fxml.MethodHelper.invoke(MethodHelper.java:84)  
    at javafx.fxml.FXMLLoader$MethodHandler.invoke(FXMLLoader.java:1860)  
    at javafx.fxml.FXMLLoader$ControllerMethodEventHandler.handle(FXMLLoader.java:1731)  
    at com.sun.javafx.event.CompositeEventHandler.dispatchBubblingEvent(CompositeEventHandler.java:86)  
    at com.sun.javafx.event.EventHandlerManager.dispatchBubblingEvent(EventHandlerManager.java:232)  
    at com.sun.javafx.event.EventHandlerManager.dispatchBubblingEvent(EventHandlerManager.java:189)  
    at com.sun.javafx.event.CompositeEventDispatcher.dispatchBubblingEvent(CompositeEventDispatcher.java:59)  
    at com.sun.javafx.event.BasicEventDispatcher.dispatchEvent(BasicEventDispatcher.java:58)  
    at com.sun.javafx.event.EventDispatchChainImpl.dispatchEvent(EventDispatchChainImpl.java:114)  
    at com.sun.javafx.event.BasicEventDispatcher.dispatchEvent(BasicEventDispatcher.java:56)
```

## Mensaje de error generado por Alertas.class

Es un error creado por una clase JavaFX, con esto queremos decir que solo se ve si el programa esta ejecutándose con una ventana activa.

Si estamos depurando el DAO no podemos usar esa clase.



## Mensajes de información de Alertas.class

Otros mensajes, más inócuos son los de tipo `Alertas.informacion`. Pueden ser de confirmación de una acción o de aviso de la no ejecución de la acción. De todos modos no indican ningún fallo grave

