

# 1 ADIMDE

Estructura de los fuentes de la aplicación.

## 1.1 *sql/torque*

app-schema.sql

Archivo sql que contiene la estructura de la base de datos.

# 2 SRC

Fuentes de la aplicación

## 2.1 *net.idtoki.adimde*

Código en JAVA de la lógica de la aplicación.

### 2.1.1 app/config

app-config.properties

Fichero con los parámetros de configuración

#

app.torque.config.file=/WEB-INF/config/app-torque.properties

app.logger.config.file=/WEB-INF/config/app-loggers.properties

# si el archivo del logger anterior no se puede cargar utiliza el indicado en

fff.logger.config.file.error

app.logger.config.file.error = ./log.log

# app.torque.database.name = z-web-app

# app.elementos.por.pagina=8

app.alias.estatico=/app

net.zylk.user.security.offline = 0

net.zylk.user.security.loginServlet =

# Sistema de seguridad

net.zylk.user.security.profilesFile=/WEB-INF/security/profiles

net.zylk.user.security.usersFile=/WEB-INF/security/users

#Localizacion de las xsl para los pdfs (fo)

app.xsl.templates.dir=c:\\adimde-app\\xsl

### 2.1.2 controller

Controladores que hacen la funcionalidad y la conexión con la parte WEB

### 2.1.3 group

Clases para la gestión de agrupación de registros.

### 2.1.4 helper

Clases de ayuda para trabajo interno.

### 2.1.5 manager

Clases para la gestión de las clases con la base de datos.

### 2.1.6 model

Clases del modelo de datos de las entidades de la base de datos.

## 3 schema

app-schema.xml

xml que define la estructura de la base de datos, Se utiliza en el proceso de auto generación de las clases (proyecto torque de apache)

## 4 data

Contiene los archivos de datos con los datos necesarios para el funcionamiento de la aplicación

turnos.txt Los diferentes turnos predefinidos

estadosproduccion.txt Los diferentes estados de la producción predefinidos

localidadesSort.txt Localidades de España

provincias.txt Provincias de España

postgresInstall.bat, adimde-data.sql y adimde-passwd Ficheros para introducción de los datos en la BBDD.

## 5 build.xml

build.xml para la construcción del war del aplicativo desde su código fuente. Se puede además utilizar para ejecutar las tareas de auto generación de código

## 6 build.properties

Definición de las características y configuración del torque-apache para la generación del código de acceso a base de datos

## 7 web

Contenido estático y jsp de la aplicación

### **7.1 WEB-INF**

### 7.1.1 config

#### ***app-loggers.properties***

archivo de configuración de los loggers. Se utiliza la clase java.util.Logger presente en las versiones de del jdk >= 1.4

```
# Sólo consola
#handlers= java.util.logging.ConsoleHandler
# Consola y fichero
handlers= java.util.logging.FileHandler, java.util.logging.ConsoleHandler
#####
# Niveles de traza
# .level= ALL
# .level= FINEST
# .level= FINER
# .level= FINE
# .level= CONFIG
# .level= INFO
# .level= WARNING
# .level= SEVERE
# .level= OFF
#####
# Máximo nivel a trazar, salvo indicación más precisa
.level= INFO
#####
# En consola
#####
java.util.logging.ConsoleHandler.level = ALL
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
#####
# En fichero
#####
java.util.logging.FileHandler.pattern = ./app.log
java.util.logging.FileHandler.level = ALL
java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
#####
# Nivel específico de un paquete [y clase]
#####
net.zylk.tools.pdf.PdfUtils.level = ALL
```

#### ***app-torque.properties***

```
#log4j.rootCategory = DEBUG, default
log4j.logger.org.apache.torque.util.BasePeer = DEBUG
log4j.appender.default = org.apache.log4j.FileAppender
log4j.appender.default.file = ./torque.log
log4j.appender.default.layout = org.apache.log4j.SimpleLayout
```

```
#####PARA POSTGRESQL#####
torque.database.default=ADIMDE-DB
```

```
torque.database.ADIMDE-DB.adapter=postgresql
torque.database.ADIMDE-DB.driver = org.postgresql.Driver
torque.database.ADIMDE-DB.url = jdbc:postgresql://localhost:5432/ADIMDE-DB
torque.database.ADIMDE-DB.username = adimdeUser
torque.database.ADIMDE-DB.password = adimdeUser
torque.dsfactory.ADIMDE-DB.factory =
org.apache.torque.dsfactory.SharedPoolDataSourceFactory
torque.dsfactory.ADIMDE-DB.pool.defaultMaxActive=10
torque.dsfactory.ADIMDE-DB.testOnBorrow=true
torque.dsfactory.ADIMDE-DB.validationQuery=SELECT 1
torque.dsfactory.ADIMDE-DB.connection.driver = org.postgresql.Driver
torque.dsfactory.ADIMDE-DB.connection.url =
jdbc:postgresql://localhost:5432/ADIMDE-DB
torque.dsfactory.ADIMDE-DB.connection.user = adimdeUser
torque.dsfactory.ADIMDE-DB.connection.password = adimdeUser
```

## 8 VENDOR

### 8.1 APP

Plantillas de Torque para la generación del código en compilación. Apuntar a estas plantillas al configurar el Torque-gen 3.1.

### 8.2 lib

librerías de la aplicación en run-time

```
fop
batik.jar
fop.jar
logkit-1.0.jar
jdbc
postgresql-8.0-311.jdbc3.jar
torque
ant-1.5.1.jar
avalon-framework-4.1.4.jar
commons-beanutils-1.6.1.jar
commons-collections-3.0.jar
commons-configuration-1.0.jar
commons-dbcp-1.2.1.jar
commons-lang-2.0.jar
commons-logging-1.0.4.jar
commons-pool-1.2.jar
jcs-20030822.182132.jar
jdbc-2.0.jar
jndi-1.2.1.jar
junit-3.8.1.jar
log4j-1.2.8.jar
logkit-1.0.1.jar
```

stratum-1.0-b5.jar  
torque-3.1.1.jar  
velocity-1.3.1.jar  
village-2.0-dev-20030825.jar  
xercesImpl-2.4.0.jar  
xml-apis-1.0.b2.jar  
xmlParserAPIs-2.0.2.jar

## 9 instalable-win32

Directorio donde se almacenan los archivos

### 9.1 xsl

xsls para la generación de los pdfs contiene los archivos xsl y xml con la información necesaria para generar los pdfs. Tiene a su vez una carpeta imágenes con un gif que se utiliza para las cabeceras de los documentos. El archivo cabecera.xml contiene los datos que se mostraran en la cabecera. El archivo main.xsl contiene las rutas absolutas de la imagen y del fichero cabecera.xml.

```
<fo:external-graphic src="file:RUTA_AL_ARCHIVO\imagen.gif"/>  
<xsl:apply-templates select="document('RUTA_AL_ARCHIVO  
\cabecera.xml')/empresa"/>
```

## 10 dist

directorio que contiene el war de la aplicacion

## 11 zlibs

directorio con librerías auxiliares