

Contents

Description

Packages

# Module com.example.interfazconodoo

module com.example.interfazconodoo

## Packages

### Exports

Package	Description
com.example.interfazconodoo	
com.example.interfazconodoo.controlad	
com.example.interfazconodoo.dao	
com.example.interfazconodoo.modelos	



Contents

Description

Related Packages

Classes and Interfaces

# Package

## com.example.interfazconodoo

package com.example.interfazconodoo

### Related Packages

Package	Description
com.example.interfazconodoo.controladore	
com.example.interfazconodoo.dao	
com.example.interfazconodoo.modelos	

### Classes

Class	Description
HelloApplication	Clase principal que inicia la aplicación JavaFX.



Description

Nested Class Summary

Field Summary

Constructor Summary

Method Summary

Constructor Details

    HelloApplication()

Method Details

    start(Stage)

    main(String[])

# Class HelloApplication

java.lang.Object  
    javafx.application.Application  
        com.example.interfazconodoo>HelloApplication

public class **HelloApplication**  
extends javafx.application.Application

Clase principal que inicia la aplicación JavaFX.

Esta clase extiende de Application y se encarga de cargar la interfaz gráfica definida en un archivo FXML, aplicar estilos CSS y mostrar la ventana principal.

## Nested Class Summary

**Nested classes/interfaces inherited from class javafx.application.Application**

javafx.application.Application.Parameters

## Field Summary

**Fields inherited from class javafx.application.Application**

STYLE SHEET\_CASPIAN, STYLE SHEET\_MODENA

## Constructor Summary

### Constructors

Constructor	Description
<a href="#">HelloApplication()</a>	

## Method Summary

- All MethodsStatic Methods
- Instance MethodsConcrete Methods

Modifier and Type	Method	Description
-------------------	--------	-------------

static void	<b>main</b> (String [] args)	Método principal que inicia la ejecución de la aplicación.
void	<b>start</b> (javafx.stage.Stage stage)	Método sobrescrito de Application que se ejecuta al iniciar la aplicación.

### Methods inherited from class `javafx.application.Application`

`getHostServices`, `getParameters`, `getUserAgentStylesheet`, `init`, `launch`, `launch`, `notifyPreloader`, `setUserAgentStylesheet`, `stop`

### Methods inherited from class `java.lang.Object`

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

## Constructor Details

### HelloApplication

```
public HelloApplication()
```

## Method Details

### start

```
public void start(javafx.stage.Stage stage)
               throws IOException
```

Método sobrescrito de Application que se ejecuta al iniciar la aplicación.

Carga la interfaz gráfica desde un archivo FXML, aplica una hoja de estilos CSS, configura el título de la ventana y la muestra en pantalla.

#### Specified by:

start in class `javafx.application.Application`

**Parameters:**

stage - La ventana principal de la aplicación.

**Throws:**

`IOException` - Si ocurre un error al cargar el archivo FXML.

**main**

```
public static void main(String [] args)
```

Método principal que inicia la ejecución de la aplicación.

Llama al método `launch()` de `Application`, el cual gestiona el ciclo de vida de la aplicación JavaFX.

**Parameters:**

args - Argumentos de línea de comandos (no utilizados en esta aplicación).

Contents

Description

Related Packages

Classes and Interfaces

Package

com.example.interfazconodoo.controladores

package com.example.interfazconodoo.controladores

Related Packages

Package	Description
com.example.interfazconodoo	
com.example.interfazconodoo.dao	
com.example.interfazconodoo.modelos	

Classes

Class	Description
FormatoController	Controlador para gestionar la interfaz gráfica de creación y edición de formatos.
HelloController	Controlador para gestionar la interfaz de visualización, búsqueda, edición y eliminación de formatos de papel.



Contents 

Filter

Description

Constructor Summary

Method Summary

Constructor Details

FormatoController()

Method Details

setFormatoNuevo(Paperformat

initialize()

onBtCrear(ActionEvent)

onBtSalir(ActionEvent)

# Class FormatoController

java.lang.Object  
com.example.interfazconodoo.controladores.FormatoConti

```
public class FormatoController
extends Object
```

Controlador para gestionar la interfaz gráfica de creación y edición de formatos.

Esta clase permite cargar, modificar y almacenar información sobre formatos de papel, gestionando los datos ingresados por el usuario y validando la información antes de guardarla.

Incluye bindings bidireccionales entre los campos de la interfaz y las propiedades del objeto `Paperformat`, asegurando la sincronización de datos en tiempo real.

## Constructor Summary

### Constructors

Constructor	Description
<code>FormatoController()</code>	

## Method Summary

All Methods   Instance Methods

### Concrete Methods

Modifier and Type	Method	Description
void	<code>initialize()</code>	Inicializa el controlador y configura la validación de los campos de entrada.
void	<code>onBtCrear</code> ( <code>javafx.event.Ac</code>	Guarda un nuevo formato o actualiza uno existente en la base de datos.

void	<b>onBtSalir</b> ( <code>javafx.event.Ac</code>	Cierra la ventana actual.
void	<b>setFormatoNuevo</b> ( <code>Paperformat</code> <code>pap</code>	Establece un nuevo formato de papel para su edición o visualización.

### Methods inherited from class `java.lang.Object`

`clone` , `equals` , `finalize` , `getClass` ,  
`hashCode` , `notify` , `notifyAll` , `toString` ,  
`wait` , `wait` , `wait`

## Constructor Details

### FormatoController

```
public FormatoController()
```

## Method Details

### setFormatoNuevo

```
public void setFormatoNuevo  
(Paperformat paperformat)
```

Establece un nuevo formato de papel para su edición o visualización.

#### Parameters:

`paperformat` - El objeto `Paperformat` que contiene los datos del formato.

### initialize

```
public void initialize()
```

Inicializa el controlador y configura la validación de los campos de entrada.

Desactiva el botón de creación hasta que todos los campos obligatorios tengan valores.



## onBtCrear

```
public void onBtCrear  
(javafx.event.ActionEvent actionEvent)  
    throws SQLException
```

Guarda un nuevo formato o actualiza uno existente en la base de datos.

Si el objeto `Paperformat` no tiene un ID asignado, se creará un nuevo registro. En caso contrario, se actualizará el registro existente. Se maneja la validación de los valores ingresados para evitar errores de formato.

### Parameters:

`actionEvent` - Evento de la interfaz que dispara la acción.

### Throws:

`SQLException` - Si ocurre un error durante la operación en la base de datos.

## onBtSalir

```
public void onBtSalir  
(javafx.event.ActionEvent actionEvent)
```

Cierra la ventana actual.

### Parameters:

`actionEvent` - Evento que dispara el cierre de la ventana.

Description

Constructor Summary

Method Summary

Constructor Details

    HelloController()

Method Details

    initialize()

    onBtnSearchBien(ActionEvent)

    onBtnEditarFormat(ActionEvent)

    onBtnEliminarFormat(ActionEvent)

    onBtnCrearFormat(ActionEvent)

# Class HelloController

java.lang.Object  
com.example.interfazconodoo.controladores.HelloController

public class **HelloController**  
extends **Object**

Controlador para gestionar la interfaz de visualización, búsqueda, edición y eliminación de formatos de papel.

Esta clase gestiona la tabla de datos y permite realizar operaciones como la búsqueda de formatos, edición de registros existentes, eliminación y creación de nuevos formatos.

## Constructor Summary

Constructors	
Constructor	Description
<b>HelloController()</b>	

## Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
void	<b>initialize()</b>	Inicializa la tabla de datos.
void	<b>onBtnCrearForma</b> (javafx.event.Ac	Abre la ventana para crear un nuevo formato.
void	<b>onBtnEditarForm</b> (javafx.event.Ac	Abre la ventana de edición de un formato seleccionado.
void	<b>onBtnEliminarFo</b> (javafx.event.Ac	Elimina el formato seleccionado tras confirmar la

		acción con una alerta.
void	<b>onBtnSearchBien</b> (javafx.event.Ac	Realiza una búsqueda de formatos por nombre y actualiza la tabla con los resultados.

**Methods inherited from class java.lang.Object**

clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

**Constructor Details**

**HelloController**

```
public HelloController()
```

**Method Details**

**initialize**

```
public void initialize()
```

Inicializa la tabla de datos.

Binding reactivo para desactivar btEliminar y btEditar si no hay selección.

Detectar cambios en el campo de búsqueda y aplicar el filtro reactivo.

**onBtnSearchBien**

```
public void onBtnSearchBien  
(javafx.event.ActionEvent actionEvent)
```

Realiza una búsqueda de formatos por nombre y actualiza la tabla con los resultados.

**Parameters:**

actionEvent - Evento que dispara la búsqueda.

### onbBtnEditarFormat

```
public void onbBtnEditarFormat  
(javafx.event.ActionEvent actionEvent)  
    throws IOException
```

Abre la ventana de edición de un formato seleccionado.

**Parameters:**

actionEvent - Evento que dispara la acción.

**Throws:**

IOException - Si hay un error al cargar la interfaz.

### onbBtnEliminarFormat

```
public void onbBtnEliminarFormat  
(javafx.event.ActionEvent actionEvent)  
    throws IOException ,  
        SQLException
```

Elimina el formato seleccionado tras confirmar la acción con una alerta.

**Parameters:**

actionEvent - Evento que dispara la acción.

**Throws:**

IOException - SQLException Si ocurre un error durante la eliminación.

SQLException

### onbBtnCrearFormat

```
public void onbBtnCrearFormat  
(javafx.event.ActionEvent actionEvent)  
    throws IOException
```

Abre la ventana para crear un nuevo formato.

**Parameters:**

actionEvent - Evento que dispara la acción.

**Throws:**

IOException - Si hay un error al cargar la interfaz.

Contents

Description

Related Packages

Classes and Interfaces

# Package

## com.example.interfazconodoo.dao

package com.example.interfazconodoo.dao

### Related Packages

Package	Description
com.example.interfazconodoo	
com.example.interfazconodoo.controladore	
com.example.interfazconodoo.modelos	

### Classes

Class	Description
Conexion	Clase para gestionar la conexión con la base de datos PostgreSQL.
Paperformatdao	Clase para gestionar la comunicación con la base de datos PostgreSQL.



Description

Field Summary

Constructor Summary

Method Summary

Field Details

connection

Constructor Details

Conexion()

Method Details

getConnection()

# Class Conexion

java.lang.Object  
com.example.interfazconodoo.dao.Conexion

public class Conexion  
extends Object

Clase para gestionar la conexión con la base de datos PostgreSQL.

Proporciona un método estático para obtener una conexión activa a la base de datos, asegurando que se inicialice solo una vez.

## Field Summary

### Fields

Modifier and Type	Field	Description
static Conexion	connection	Instancia única de la conexión a la base de datos.

## Constructor Summary

### Constructors

Constructor	Description
Conexion()	

## Method Summary

All MethodsStatic Methods

Concrete Methods

Modifier and Type	Method	Description
static Conexion	getConnection()	Obtiene la conexión a la base de datos PostgreSQL.

## Methods inherited from

**class java.lang.Object**

`clone` , `equals` , `finalize` , `getClass` ,  
`hashCode` , `notify` , `notifyAll` , `toString` ,  
`wait` , `wait` , `wait`

**Field Details****connection**

`public static Connection connection`

Instancia única de la conexión a la base de datos.

**Constructor Details****Conexion**

`public Conexion()`

**Method Details****getConnection**

`public static Connection getConnection()`

Obtiene la conexión a la base de datos PostgreSQL.

Si la conexión aún no ha sido creada, este método la inicializa. En caso de fallo, lanza una excepción en tiempo de ejecución.

**Returns:**

La conexión activa a la base de datos.

**Throws:**

`RuntimeException` - Si ocurre un error al intentar conectar a la base de datos.

Description

Constructor Summary

Method Summary

Constructor Details

Paperformatdao()

Method Details

obtenerFormatos(String)

crearFormato(Paperformat)

eliminarFormato(Paperformat)

editarFormato(Paperformat)

# Class Paperformatdao

java.lang.Object  
com.example.interfazconodoo.dao.Paperformatdao

public class **Paperformatdao**  
extends **Object**

Clase para gestionar la comunicación con la base de datos PostgreSQL.

Proporciona métodos para realizar operaciones CRUD en la tabla "report\_paperformat".

## Constructor Summary

### Constructors

Constructor	Description
<b>Paperformatdao()</b>	

## Method Summary

All Methods

Static Methods

### Concrete Methods

Modifier and Type	Method	Description
static void	<b>crearFormato</b> ( <b>Paperformat</b> for	Inserta un nuevo formato en la base de datos.
static void	<b>editarFormato</b> ( <b>Paperformat</b> for	Actualiza un formato existente en la base de datos.
static void	<b>eliminarFormato</b> ( <b>Paperformat</b> for	Elimina un formato de la base de datos según su ID.
static <b>List</b> < <b>Paperformat</b> >	<b>obtenerFormatos</b> ( <b>String</b> nombre)	Obtiene los formatos almacenados en la base de datos que coincidan



con el nombre  
proporcionado.

### Methods inherited from class `java.lang.Object`

`clone` , `equals` , `finalize` , `getClass` ,  
`hashCode` , `notify` , `notifyAll` , `toString` ,  
`wait` , `wait` , `wait`

## Constructor Details

### Paperformatdao

```
public Paperformatdao()
```

## Method Details

### obtenerFormatos

```
public static List  
<Paperformat> obtenerFormatos(String nombre)  
                                throws  
SQLException
```

Obtiene los formatos almacenados en la base de datos que coincidan con el nombre proporcionado.

#### Parameters:

nombre - Nombre parcial o completo del formato a buscar.

#### Returns:

Lista de objetos `Paperformat` que coinciden con el criterio de búsqueda.

#### Throws:

`SQLException` - Si ocurre un error al ejecutar la consulta.

### crearFormato

```
public static void crearFormato  
(Paperformat formato)  
                                throws SQLException
```

Inserta un nuevo formato en la base de datos.

**Parameters:**

formato - Objeto `Paperformat` a insertar.

**Throws:**

`SQLException` - Si ocurre un error durante la inserción.

**eliminarFormato**

```
public static void eliminarFormato  
(Paperformat formato)
```

throws

`SQLException`

Elimina un formato de la base de datos según su ID.

**Parameters:**

formato - Objeto `Paperformat` que se eliminará.

**Throws:**

`SQLException` - Si ocurre un error durante la eliminación.

**editarFormato**

```
public static void editarFormato  
(Paperformat formato)
```

throws `SQLException`

Actualiza un formato existente en la base de datos.

**Parameters:**

formato - Objeto `Paperformat` con los datos actualizados.

**Throws:**

`SQLException` - Si ocurre un error durante la actualización.

Contents

Description

Related Packages

Classes and Interfaces

# Package

## com.example.interfazconodoo.modelos

package com.example.interfazconodoo.modelos

### Related Packages

Package	Description
<a href="#">com.example.interfazconodoo</a>	
<a href="#">com.example.interfazconodoo.controladore</a>	
<a href="#">com.example.interfazconodoo.dao</a>	

### Classes

Class	Description
<a href="#">Paperformat</a>	Clase que representa la estructura de la tabla "report_paperformat" en la base de datos.



Contents

Filter

Description

Property Summary

Constructor Summary

Method Summary

Property Details

id

pageHeight

pageWidth

headerSpacing

dpi

createUid

writeUid

name

format

orientation

default

headerLine

disableShrinking

createDate

writeDate

marginTop

marginBottom

marginLeft

marginRight

Constructor Details

Paperformat(Integer, String, String, String, Double, Double, Double, Double)

Panerformat()

Class Paperformat

java.lang.Object  
com.example.interfazconodoo.modelos.Paperformat

public class Paperformat  
extends Object

Clase que representa la estructura de la tabla "report\_paperformat" en la base de datos. Contiene propiedades con sus respectivos métodos de acceso y modificación.

Esta clase utiliza propiedades de JavaFX para facilitar la vinculación con la interfaz gráfica.

Property Summary

Properties

Type	Property	Description
javafx.beans.property. <Timestamp >	createDate	
javafx.beans.property.	createUid	
javafx.beans.property.	default	
javafx.beans.property.	disableShrinking	
javafx.beans.property.	dpi	
javafx.beans.property.	format	Obtiene la propiedad del formato del papel.
javafx.beans.property.	headerLine	
javafx.beans.property.	headerSpacing	
javafx.beans.property.	id	
javafx.beans.property.	marginBottom	Obtiene la propiedad del margen inferior.
javafx.beans.property.	marginLeft	Obtiene la propiedad del

	margen izquierdo.
<code>javafx.beans.p marginRight</code>	Obtiene la propiedad del margen derecho.
<code>javafx.beans.p marginTop</code>	Obtiene la propiedad del margen superior.
<code>javafx.beans.p name</code>	Obtiene la propiedad name, útil para enlazarla con UI en JavaFX.
<code>javafx.beans.p orientation</code>	Obtiene la propiedad de la orientación del papel.
<code>javafx.beans.p pageHeight</code>	
<code>javafx.beans.p pageWidth</code>	
<code>javafx.beans.p writeDate &lt;Timestamp &gt;</code>	
<code>javafx.beans.p writeUid</code>	

Constructor Summary

Constructors

Constructor	Description
<code>Paperformat()</code>	Constructor vacio.
<code>Paperformat(Integer id, String name, String format, String orientation, Double marginTop, Double marginBottom, Double marginLeft, Double marginRight)</code>	Constructor con parámetros.

Method Summary

All Methods		Instance Methods
Concrete Methods		
Modifier and Type	Method	Description
javafx.beans.property<Timestamp>	createDateProperty()	
javafx.beans.property	createUidProperty()	
javafx.beans.property	defaultProperty()	
javafx.beans.property	disableShrinking()	
javafx.beans.property	dpiProperty()	
javafx.beans.property	formatProperty()	Obtiene la propiedad del formato del papel.
Timestamp	getCreateDate()	Gets the value of the createDate property.
int	getCreateUid()	Gets the value of the createUid property.
int	getDpi()	Gets the value of the dpi property.
String	getFormat()	Obtiene el tipo de formato de la página (Ejemplo: A4, Letter, etc.).
int	getHeaderSpacing()	Gets the value of the headerSpacing property.
int	getId()	Obtiene el identificador del formato.
double	getMarginBottom()	Obtiene el margen inferior.
double	getMarginLeft()	Obtiene el margen izquierdo.

double	<b>getMarginRight()</b>	Obtiene el margen derecho.
double	<b>getMarginTop()</b>	Obtiene el margen superior.
<b>String</b>	<b>getName()</b>	Obtiene el nombre del formato.
<b>String</b>	<b>getOrientation()</b>	Obtiene la orientación del papel (Portrait o Landscape).
int	<b>getPageHeight()</b>	Gets the value of the pageHeight property.
int	<b>getPageWidth()</b>	Gets the value of the pageWidth property.
<b>Timestamp</b>	<b>getWriteDate()</b>	Gets the value of the writeDate property.
int	<b>getWriteUid()</b>	Gets the value of the writeUid property.
javafx.beans.property headerLineProperty		
javafx.beans.property headerSpacingProperty		
javafx.beans.property idProperty()		
boolean	<b>isDefault()</b>	Gets the value of the default property.
boolean	<b>isDisableShrinking()</b>	Gets the value of the disableShrinking property.
boolean	<b>isHeaderLine()</b>	Gets the value of the headerLine property.
javafx.beans.property marginBottomProperty		

`javafx.beans.property` **marginLeftProperty()** Obtiene la propiedad del margen izquierdo.

`javafx.beans.property` **marginRightProperty()** Obtiene la propiedad del margen derecho.

`javafx.beans.property` **marginTopProperty()** Obtiene la propiedad del margen superior.

`javafx.beans.property` **nameProperty()** Obtiene la propiedad name, útil para enlazarla con UI en JavaFX.

`javafx.beans.property` **orientationProperty()** Obtiene la propiedad de la orientación del papel.

`javafx.beans.property` **pageHeightProperty()**

`javafx.beans.property` **pageWidthProperty()**

`void` **setCreateDate(Timestamp createDate)** Sets the value of the createDate property.

`void` **setCreateUid(int createUid)** Sets the value of the createUid property.

`void` **setDefault(boolean defecto)** Sets the value of the default property.

`void` **setDisableShrink(boolean disableShrink)** Sets the value of the disableShrink property.

`void` **setDpi(int dpi)** Sets the value of the dpi property.

`void` **setFormat(String format)** Establece el tipo de formato del papel.

`void` **setHeaderLine(boolean headerLine)** Sets the value of the headerLine property.



		property.
void	<b>setHeaderSpacing</b> (int headerSpaci	Sets the value of the headerSpacing property.
void	<b>setId</b> (int id)	Establece el identificador del formato.
void	<b>setMarginBottom</b> (double marginBo	Establece el margen inferior.
void	<b>setMarginLeft</b> (double marginLe	Establece el margen izquierdo.
void	<b>setMarginRight</b> (double marginRi	Establece el margen derecho.
void	<b>setMarginTop</b> (double marginTo	Establece el margen superior.
void	<b>setName</b> (String name)	Establece el nombre del formato.
void	<b>setOrientation</b> (String orientation)	Establece la orientación del papel.
void	<b>setPageHeight</b> (int pageHeight)	Sets the value of the pageHeight property.
void	<b>setPageWidth</b> (int pageWidth)	Sets the value of the pageWidth property.
void	<b>setWriteDate</b> (Timestamp writeDate)	Sets the value of the writeDate property.
void	<b>setWriteUid</b> (int writeUid)	Sets the value of the writeUid property.

javafx.beans.p **writeDateProperty**  
<Timestamp >

javafx.beans.p **writeUidProperty**

## Methods inherited from class `java.lang.Object`

`clone` , `equals` , `finalize` , `getClass` ,  
`hashCode` , `notify` , `notifyAll` , `toString` ,  
`wait` , `wait` , `wait`

## Property Details

### id

```
public javafx.beans.property.IntegerProperty id
```

#### See Also:

```
getId(), setId(int), idProperty()
```



### pageHeight

```
public javafx.beans.property.IntegerProperty pageHeight
```

#### See Also:

```
getPageHeight(), setPageHeight(int),  
pageHeightProperty()
```



### pageWidth

```
public javafx.beans.property.IntegerProperty pageWidth
```

#### See Also:

```
getPageWidth(), setPageWidth(int),  
pageWidthProperty()
```



### headerSpacing

```
public javafx.beans.property.IntegerProperty headerSpacing
```

#### See Also:

```
getHeaderSpacing(), setHeaderSpacing(int),  
headerSpacingProperty()
```



### dpi

```
public javafx.beans.property.IntegerProperty dpi
```

**See Also:**

```
getDpi(), setDpi(int), dpiProperty()
```

**createUid**

```
public javafx.beans.property.IntegerProperty createUid
```

**See Also:**

```
getCreateUid(), setCreateUid(int),  
createUidProperty()
```

**writeUid**

```
public javafx.beans.property.IntegerProperty writeUid
```

**See Also:**

```
getWriteUid(), setWriteUid(int),  
writeUidProperty()
```

**name**

```
public javafx.beans.property.StringProperty name
```

Obtiene la propiedad name, útil para enlazarla con UI en JavaFX.

**See Also:**

```
getName(), setName(String), nameProperty()
```

**format**

```
public javafx.beans.property.StringProperty format
```

Obtiene la propiedad del formato del papel.

**See Also:**

```
getFormat(), setFormat(String),  
formatProperty()
```

**orientation**

```
public javafx.beans.property.StringProperty orientationProperty()
```

Obtiene la propiedad de la orientación del papel.

**See Also:**

```
getOrientation(), setOrientation(String),  
orientationProperty()
```

**default**

```
public javafx.beans.property.BooleanProperty defaultProperty()
```

**See Also:**

```
isDefault(), setDefault(boolean),  
defaultProperty()
```

**headerLine**

```
public javafx.beans.property.BooleanProperty headerLineProperty()
```

**See Also:**

```
isHeaderLine(), setHeaderLine(boolean),  
headerLineProperty()
```

**disableShrinking**

```
public javafx.beans.property.BooleanProperty disableShrinkingProperty()
```

**See Also:**

```
isDisableShrinking(),  
setDisableShrinking(boolean),  
disableShrinkingProperty()
```

**createDate**

```
public javafx.beans.property.ObjectProperty<Timestamp>  
createDateProperty()
```

**See Also:**

```
getCreateDate(), setCreateDate(Timestamp),  
createDateProperty()
```



## writeDate

```
public javafx.beans.property.ObjectProperty<Timestamp> writeDateProperty
```

### See Also:

```
getWriteDate(), setWriteDate(Timestamp),  
writeDateProperty()
```



## marginTop

```
public javafx.beans.property.DoubleProperty marginTopProperty
```

Obtiene la propiedad del margen superior.

### See Also:

```
getMarginTop(), setMarginTop(double),  
marginTopProperty()
```



## marginBottom

```
public javafx.beans.property.DoubleProperty marginBottomProperty
```

Obtiene la propiedad del margen inferior.

### See Also:

```
getMarginBottom(), setMarginBottom(double),  
marginBottomProperty()
```



## marginLeft

```
public javafx.beans.property.DoubleProperty marginLeftProperty
```

Obtiene la propiedad del margen izquierdo.

### See Also:

```
getMarginLeft(), setMarginLeft(double),  
marginLeftProperty()
```



## marginRight

```
public javafx.beans.property.DoubleProperty marginRightProperty
```

Obtiene la propiedad del margen derecho.

**See Also:**

```
getMarginRight(), setMarginRight(double),  
marginRightProperty()
```

## Constructor Details

### Paperformat

```
public Paperformat(Integer id,  
                    String name,  
                    String format,  
                    String orientation,  
                    Double marginTop,  
                    Double marginBottom,  
                    Double marginLeft,  
                    Double marginRight)
```

Constructor con parámetros.

**Parameters:**

id - Identificador del formato.

name - Nombre del formato.

format - Formato de la página (Ejemplo: A4, Letter, etc.).

orientation - Orientación del formato (Portrait o Landscape).

marginTop - Margen superior.

marginBottom - Margen inferior.

marginLeft - Margen izquierdo.

marginRight - Margen derecho.

### Paperformat

```
public Paperformat()
```

Constructor vacío.

## Method Details

### idProperty

```
public javafx.beans.property.IntegerProperty idl
```

**Returns:**

the id property

**See Also:**

`getId()`, `setId(int)`

**getId**

```
public int getId()
```

Obtiene el identificador del formato.

**Returns:**

ID del formato.

**setId**

```
public void setId(int id)
```

Establece el identificador del formato.

**Parameters:**

id - Nuevo ID del formato.

**pageHeightProperty**

```
public javafx.beans.property.IntegerProperty pageHeightProperty()
```

**Returns:**

the pageHeight property

**See Also:**

`getPageHeight()`, `setPageHeight(int)`

**getPageHeight**

```
public int getPageHeight()
```

Gets the value of the pageHeight property.

**Property description:****Returns:**

the value of the pageHeight property

**See Also:**

`setPageHeight(int)`, `pageHeightProperty()`

## setPageHeight

```
public void setPageHeight(int pageHeight)
```

Sets the value of the pageHeight property.

**Property description:****Parameters:**

pageHeight - the value for the pageHeight property

**See Also:**

[getPageHeight\(\)](#), [pageHeightProperty\(\)](#)

## pageWidthProperty

```
public javafx.beans.property.IntegerProperty pageWidthProperty()
```

**Returns:**

the pageWidth property

**See Also:**

[getPageWidth\(\)](#), [setPageWidth\(int\)](#)

## getPageWidth

```
public int getPageWidth()
```

Gets the value of the pageWidth property.

**Property description:****Returns:**

the value of the pageWidth property

**See Also:**

[setPageWidth\(int\)](#), [pageWidthProperty\(\)](#)

## setPageWidth

```
public void setPageWidth(int pageWidth)
```

Sets the value of the pageWidth property.

**Property description:****Parameters:**

pageWidth - the value for the pageWidth property

**See Also:**

[getPageWidth\(\)](#), [pageWidthProperty\(\)](#)



## headerSpacingProperty

```
public javafx.beans.property.IntegerProperty headerSpacingProperty()
```

**Returns:**

the headerSpacing property

**See Also:**

[getHeaderSpacing\(\)](#), [setHeaderSpacing\(int\)](#)



## getHeaderSpacing

```
public int getHeaderSpacing()
```

Gets the value of the headerSpacing property.

**Property description:****Returns:**

the value of the headerSpacing property

**See Also:**

[setHeaderSpacing\(int\)](#),  
[headerSpacingProperty\(\)](#)

## setHeaderSpacing

```
public void setHeaderSpacing(int headerSpacing)
```

Sets the value of the headerSpacing property.

**Property description:****Parameters:**

headerSpacing - the value for the headerSpacing property

**See Also:**

[getHeaderSpacing\(\)](#), [headerSpacingProperty\(\)](#)

## dpiProperty

```
public javafx.beans.property.IntegerProperty dpiProperty()
```

**Returns:**

the dpi property

**See Also:**

[getDpi\(\)](#), [setDpi\(int\)](#)



## getDpi

```
public int getDpi()
```

Gets the value of the dpi property.

**Property description:****Returns:**

the value of the dpi property

**See Also:**

`setDpi(int)`, `dpiProperty()`

## setDpi

```
public void setDpi(int dpi)
```

Sets the value of the dpi property.

**Property description:****Parameters:**

dpi - the value for the dpi property

**See Also:**

`getDpi()`, `dpiProperty()`

## createUidProperty

```
public javafx.beans.property.IntegerProperty createUidProperty()
```

**Returns:**

the createUid property

**See Also:**

`getCreateUid()`, `setCreateUid(int)`



## getCreateUid

```
public int getCreateUid()
```

Gets the value of the createUid property.

**Property description:****Returns:**

the value of the createUid property

**See Also:**

`setCreateUid(int)`, `createUidProperty()`

## setCreateUid

```
public void setCreateUid(int createUid)
```

Sets the value of the createUid property.

**Property description:****Parameters:**

createUid - the value for the createUid property

**See Also:**

`getCreateUid()`, `createUidProperty()`

## writeUidProperty

```
public javafx.beans.property.IntegerProperty writeUidProperty()
```

**Returns:**

the writeUid property

**See Also:**

`getWriteUid()`, `setWriteUid(int)`

## getWriteUid

```
public int getWriteUid()
```

Gets the value of the writeUid property.

**Property description:****Returns:**

the value of the writeUid property

**See Also:**

`setWriteUid(int)`, `writeUidProperty()`

## setWriteUid

```
public void setWriteUid(int writeUid)
```

Sets the value of the writeUid property.

**Property description:****Parameters:**

writeUid - the value for the writeUid property

**See Also:**

`getWriteUid()`, `writeUidProperty()`

## nameProperty

```
public javafx.beans.property.StringProperty nameProperty()
```

Obtiene la propiedad name, útil para enlazarla con UI en JavaFX.

**Returns:**

Propiedad del nombre del formato.

**See Also:**

`getName()`, `setName(String)`



## getName

```
public String getName()
```

Obtiene el nombre del formato.

**Returns:**

Nombre del formato.

## setName

```
public void setName(String name)
```

Establece el nombre del formato.

**Parameters:**

name - Nuevo nombre del formato.

## formatProperty

```
public javafx.beans.property.StringProperty formatProperty()
```

Obtiene la propiedad del formato del papel.

**Returns:**

Propiedad del formato.

**See Also:**

`getFormat()`, `setFormat(String)`



## getFormat

```
public String getFormat()
```

Obtiene el tipo de formato de la página (Ejemplo: A4, Letter, etc.).

**Returns:**

Formato del papel.

**setFormat**

```
public void setFormat(String format)
```

Establece el tipo de formato del papel.

**Parameters:**

format - Nuevo tipo de formato del papel.

**orientationProperty**

```
public javafx.beans.property.StringProperty orientationProperty()
```

Obtiene la propiedad de la orientación del papel.

**Returns:**

Propiedad de la orientación.

**See Also:**

`getOrientation()`, `setOrientation(String)`

**getOrientation**

```
public String getOrientation()
```

Obtiene la orientación del papel (Portrait o Landscape).

**Returns:**

Orientación del formato.

**setOrientation**

```
public void setOrientation(String orientation)
```

Establece la orientación del papel.

**Parameters:**

orientation - Nueva orientación del formato.

**defaultProperty**

```
public javafx.beans.property.BooleanProperty de
```

**Returns:**

the default property

**See Also:**

[isDefault\(\)](#), [setDefault\(boolean\)](#)



## isDefault

```
public boolean isDefault()
```

Gets the value of the default property.

**Property description:****Returns:**

the value of the default property

**See Also:**

[setDefault\(boolean\)](#), [defaultProperty\(\)](#)

## setDefault

```
public void setDefault(boolean defecto)
```

Sets the value of the default property.

**Property description:****Parameters:**

defecto - the value for the default property

**See Also:**

[isDefault\(\)](#), [defaultProperty\(\)](#)

## headerLineProperty

```
public javafx.beans.property.BooleanProperty he
```

**Returns:**

the headerLine property

**See Also:**

[isHeaderLine\(\)](#), [setHeaderLine\(boolean\)](#)



## isHeaderLine

```
public boolean isHeaderLine()
```

Gets the value of the headerLine property.

**Property description:****Returns:**

the value of the headerLine property

**See Also:**

`setHeaderLine(boolean)`, `headerLineProperty()`

## setHeaderLine

```
public void setHeaderLine(boolean headerLine)
```

Sets the value of the headerLine property.

**Property description:****Parameters:**

headerLine - the value for the headerLine property

**See Also:**

`isHeaderLine()`, `headerLineProperty()`

## disableShrinkingProperty

```
public javafx.beans.property.BooleanProperty disableShrinkingProperty()
```

**Returns:**

the disableShrinking property

**See Also:**

`isDisableShrinking()`,  
`setDisableShrinking(boolean)`



## isDisableShrinking

```
public boolean isDisableShrinking()
```

Gets the value of the disableShrinking property.

**Property description:****Returns:**

the value of the disableShrinking property

**See Also:**

`setDisableShrinking(boolean)`,  
`disableShrinkingProperty()`

## setDisableShrinking

```
public void setDisableShrinking  
(boolean disableShrinking)
```

Sets the value of the disableShrinking property.

**Property description:****Parameters:**

disableShrinking - the value for the  
disableShrinking property

**See Also:**

`isDisableShrinking()`,  
`disableShrinkingProperty()`

## createDateProperty

```
public javafx.beans.property.ObjectProperty<Time  
> createDateProperty()
```

**Returns:**

the createDate property

**See Also:**

`getCreateDate()`, `setCreateDate(Timestamp)`



## getCreateDate

```
public Timestamp getCreateDate()
```

Gets the value of the createDate property.

**Property description:****Returns:**

the value of the createDate property

**See Also:**

`setCreateDate(Timestamp)`,  
`createDateProperty()`

## setCreateDate

```
public void setCreateDate  
(Timestamp createDate)
```

Sets the value of the createDate property.

**Property description:****Parameters:**

createDate - the value for the createDate property



**See Also:**

`getCreateDate()`, `createDateProperty()`

**writeDateProperty**

```
public javafx.beans.property.ObjectProperty<Timestamp> writeDateProperty()
```

**Returns:**

the writeDate property

**See Also:**

`getWriteDate()`, `setWriteDate(Timestamp)`

**getWriteDate**

```
public Timestamp getWriteDate()
```

Gets the value of the writeDate property.

**Property description:****Returns:**

the value of the writeDate property

**See Also:**

`setWriteDate(Timestamp)`, `writeDateProperty()`

**setWriteDate**

```
public void setWriteDate(Timestamp writeDate)
```

Sets the value of the writeDate property.

**Property description:****Parameters:**

writeDate - the value for the writeDate property

**See Also:**

`getWriteDate()`, `writeDateProperty()`

**marginTopProperty**

```
public javafx.beans.property.DoubleProperty marginTopProperty()
```

Obtiene la propiedad del margen superior.

**Returns:**

Propiedad del margen superior.

**See Also:**`getMarginTop(), setMarginTop(double)`**getMarginTop**

```
public double getMarginTop()
```

Obtiene el margen superior.

**Returns:**

Margen superior en unidades de medida de la base de datos.

**setMarginTop**

```
public void setMarginTop(double marginTop)
```

Establece el margen superior.

**Parameters:**

marginTop - Nuevo valor para el margen superior.

**marginBottomProperty**

```
public javafx.beans.property.DoubleProperty marginBottomProperty()
```

Obtiene la propiedad del margen inferior.

**Returns:**

Propiedad del margen inferior.

**See Also:**`getMarginBottom(), setMarginBottom(double)`**getMarginBottom**

```
public double getMarginBottom()
```

Obtiene el margen inferior.

**Returns:**

Margen inferior en unidades de medida de la base de datos.

**setMarginBottom**

```
public void setMarginBottom  
(double marginBottom)
```

Establece el margen inferior.

**Parameters:**

marginBottom - Nuevo valor para el margen inferior.

**marginLeftProperty**

```
public javafx.beans.property.DoubleProperty marginLeftProperty()
```

Obtiene la propiedad del margen izquierdo.

**Returns:**

Propiedad del margen izquierdo.

**See Also:**

[getMarginLeft\(\)](#), [setMarginLeft\(double\)](#)

**getMarginLeft**

```
public double getMarginLeft()
```

Obtiene el margen izquierdo.

**Returns:**

Margen izquierdo en unidades de medida de la base de datos.

**setMarginLeft**

```
public void setMarginLeft(double marginLeft)
```

Establece el margen izquierdo.

**Parameters:**

marginLeft - Nuevo valor para el margen izquierdo.

**marginRightProperty**

```
public javafx.beans.property.DoubleProperty marginRightProperty()
```

Obtiene la propiedad del margen derecho.

**Returns:**

Propiedad del margen derecho.

**See Also:**`getMarginRight(), setMarginRight(double)`**getMarginRight**

```
public double getMarginRight()
```

Obtiene el margen derecho.

**Returns:**

Margen derecho en unidades de medida de la base de datos.

**setMarginRight**

```
public void setMarginRight(double marginRight)
```

Establece el margen derecho.

**Parameters:**

`marginRight` - Nuevo valor para el margen derecho.