**USTA EMPRESARIAL**

**ESPECIFICACIÓN FUNCIONAL**

MÓDULO CRM

Versión 1.0

Enero 22 de 2016

|  |  |  |  |
| --- | --- | --- | --- |
| **REGISTRO DE CAMBIOS AL DOCUMENTO** | | | |
|  | | | |
| **Versión** | **Fecha** | **Descripción del cambio** | **Autor** |
| 1.0 | 6/05/2019 | Desarrollo del documento módulo CRM | Grupo: Los 5 furiosos |
| 1.1 | 15 mayo 2019 | Creación del Ejb de la tabla CRM | The Alirios |
| 1.1 | 26 mayo 2019 | Desarrollo de la documentación | The Alirios |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLA DE CONTENIDO**

**INTRODUCCIÓN**

**1. DESCRIPCIÓN MÓDULO CRM**

**2. BASE DE DATOS**

**3. DESCRIPCIÓN TABLAS**

**3.1. TIPO MEDIO**

**3.1.1. Atributos**

**3.1.2. Llave Primaria y Llaves Foráneas**

**3.1.3. DDL**

**3.1.4. Inserción de Datos**

**3.2. PERIODO**

**3.2.1. Atributos**

**3.2.2. Llave Primaria y Llaves Foráneas**

**3.2.3. DDL**

**3.2.4. Inserción de Datos**

**3.3. CAMPAÑA**

**3.3.1. Atributos**

**3.3.2. Llave Primaria y Llaves Foráneas**

**3.3.3. DDL**

**3.3.4. Inserción de Datos**

**3.4. EVENTO**

**3.4.1. Atributos**

**3.4.2. Llave Primaria y Llaves Foráneas**

**3.4.3. DDL**

**3.4.4. Inserción de Datos**

**3.5. RECURSO**

**3.5.1. Atributos**

**3.5.2. Llave Primaria y Llaves Foráneas**

**3.5.3. DDL**

**3.5.4. Inserción de Datos**

**3.6. MEDIO**

**3.6.1. Atributos**

**3.6.2. Llave Primaria y Llaves Foráneas**

**3.6.3. DDL**

**3.6.4. Inserción de Datos**

**INTRODUCCIÓN**

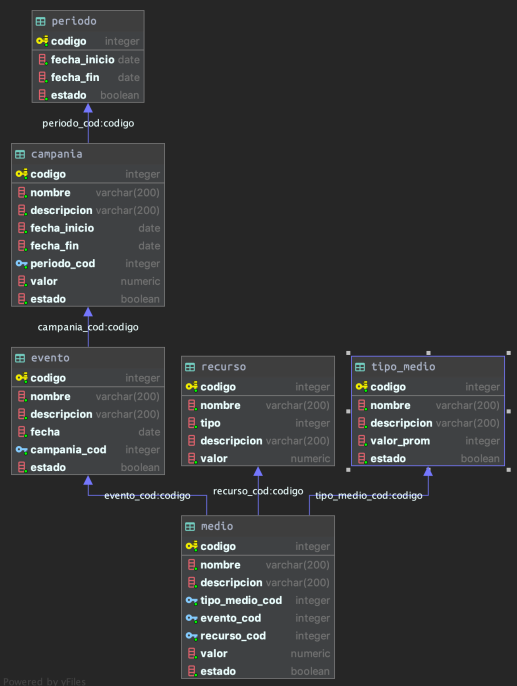
La presente documentación estará basada en el módulo CRM del proyecto USTA Empresarial. Se encontrará todo lo relacionado principalmente con la base de datos y sus descripciones en cuanto a tablas y columnas, inserción de datos y código SQL para su respectiva creación.

1. **DESCRIPCIÓN DEL MÓDULO CRM**

Es el módulo del proyecto encargado del marketing y publicidad de la empresa, el cual da conocimiento de los medios por los cuales serán expuestas las campañas existentes, los recursos que serán utilizados para las mismas, el período en el cual serán publicadas y el período en el que finalizará la publicidad de cada campaña.

1. **BASE DE DATOS**

El diseño y modelado de la base de datos del módulo CRM fue construido con PostgreSQL**.** Se encuentra en uno de los ‘schemas’ de la base de datos *usta\_empresarial* llamado ***CRM*** y cuenta con 6 tablas.

****

**Imagen1. Diagrama E-R. Módulo CRM**

1. **DESCRIPCIÓN TABLAS**

**3.1. TIPO MEDIO**

Tabla en la cual se encuentra la descripción y características de los tipos de medios publicitarios disponibles para la publicación de las campañas.

**3.1.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estarán identificados los tipos de medio de publicidad de las campañas.
* **Nombre:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se especifica el nombre del tipo de medio de publicidad.
* **Descripción:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se hace la respectiva descripción del tipo de medio de publicidad.
* **Valor Promedio:** Es un atributo obligatorio de tipo *INT* que hace referencia al precio promedio del tipo de medio de publicidad.
* **Estado:** Es un atributo obligatorio de tipo *BOOLEAN* que hace referencia al estado de cada tipo de medio de publicidad. Sus posibles estados se definen de la siguiente manera:

**3.1.2. Llave Primaria y Llaves Foráneas**

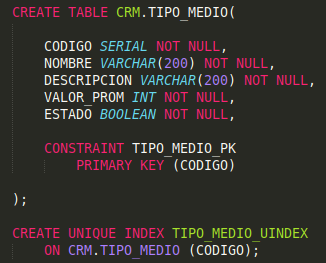
**Llave primaria:**

* **TIPO\_MEDIO\_PK:** El código es el identificador del tipo de medio publicitario, de esta manera se distinguen los tipos de medios y se identificarán sus características.

**Llaves Foráneas:**

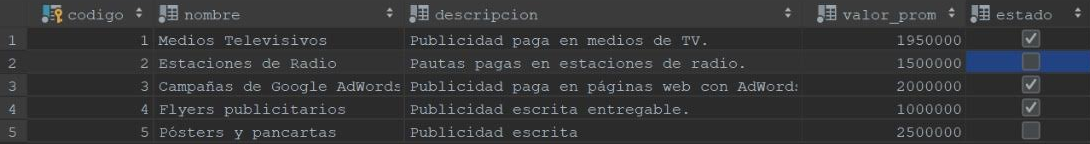
* La tabla TIPO\_MEDIO no recibe ningún dato, por el contrario, la tabla MEDIO recibe como llave foránea la columna CÓDIGO.

**3.1.3. DDL**

****

**Imagen2. DDL de la creación de la tabla TIPO\_MEDIO**

**3.1.4. Inserción de Datos**

****

**Imagen2.1. datos insertados en la tabla TIPO\_MEDIO**

**3.2. PERIODO**

Tabla en la cual se encuentra la descripción y características del periodo de publicidad de la campaña.

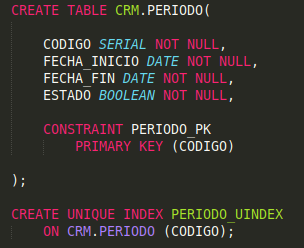
**3.2.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estará identificado cada uno de los períodos de publicidad de las campañas.
* **Fecha inicio:** Es un atributo obligatorio de tipo *DATE* que especifica la fecha de inicio del período de publicación de la campaña.
* **Fecha fin:** Es un atributo obligatorio de tipo *DATE* que especifica la fecha fin del período de publicación de la campaña.
* **Estado:** Es un atributo obligatorio de tipo *BOOLEAN* que hace referencia al estado de cada uno de los períodos de publicidad de las campañas. Sus posibles estados se definen de la siguiente manera:

**3.2.2. Llave Primaria y Llaves Foráneas**

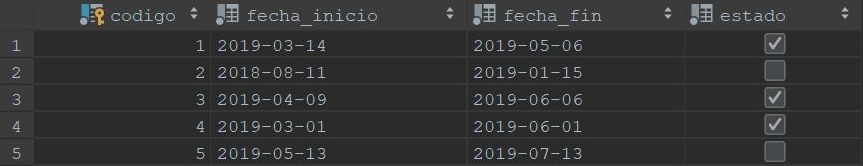
* **Llave Primaria:** La llave primaria es “código” el cual volverá único el identificador del periodo.
* **Llaves Foráneas:** No tiene llaves foráneas.

**3.2.3. DDL**

****

**Imagen3. DDL de la creación de la tabla PERIODO**

**3.2.4. Inserción de Datos**

****

**Imagen3.1. datos insertados en la tabla PERIODO**

**3.3. CAMPAÑA**

Tabla en la cual se encuentra la descripción y características de las campañas pertenecientes a la empresa.

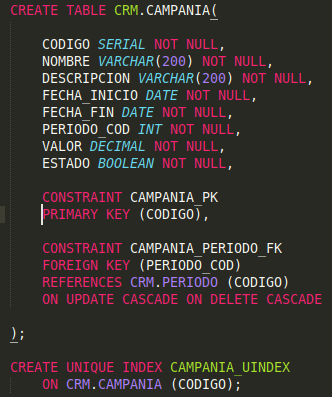
**3.3.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estará identificada cada una de las campañas publicitarias.
* **Nombre:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se especifica el nombre de la campaña publicitaria.
* **Descripción:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se hace la respectiva descripción del nombre de la campaña publicitaria.
* **Fecha inicio:** Es un atributo obligatorio de tipo *DATE* que especifica la fecha de inicio de la campaña publicitaria.
* **Fecha fin:** Es un atributo obligatorio de tipo *DATE* que especifica la fecha fin de la campaña publicitaria.
* **Código periodo:** Es un atributo heredado de la tabla PERIODO, especifica el período asignado a la campaña publicitaria.
* **Valor:** Es un atributo obligatorio de tipo *DECIMAL* que hace referencia al precio promedio de cada una de las campañas publicitarias.
* **Estado:** Es un atributo obligatorio de tipo *BOOLEAN* que hace referencia al estado de cada una de las campañas publicitarias. Sus posibles estados se definen de la siguiente manera:

**3.3.2. Llave Primaria y Llaves Foráneas**

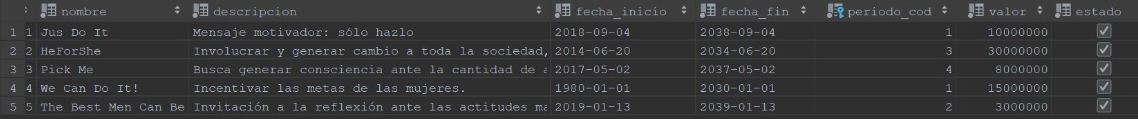
* **Llave Primaria:** La llave primaria es “código” el cual volverá único el identificador de la campaña.
* **Llaves Foráneas:** La llave foránea es “periodo\_cod” la que llegará heredada de la tabla periodo , este es el código identificador del periodo.

**3.3.3. DDL**

****

**Imagen4. DDL de la creación de la tabla CAMPAÑA**

**3.3.4. Inserción de Datos**

****

**Imagen4.1. datos insertados en la tabla CAMPAÑA**

**3.4. EVENTO**

Tabla en la cual se encuentra la descripción y características de los eventos de la empresa por los cuales serán reconocidos los servicios de la misma.

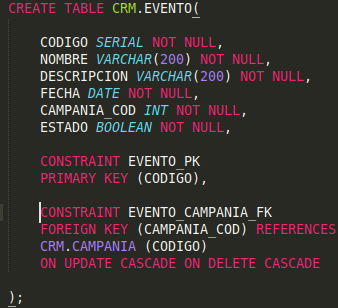
**3.4.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estará identificado cada uno de los eventos.
* **Nombre:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se especifica el nombre de cada evento.
* **Descripción:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se hace la respectiva descripción del nombre de cada evento.
* **Fecha:** Es un atributo obligatorio de tipo *DATE* que especifica la fecha del evento.
* **Código Campaña:** Es un atributo heredado de la tabla CAMPAÑA, especifica el código asignado a cada una de las campañas.
* **Estado:** Es un atributo obligatorio de tipo *BOOLEAN* que hace referencia al estado de cada uno de los eventos. Sus posibles estados se definen de la siguiente manera:

**3.4.2. Llave Primaria y Llaves Foráneas**

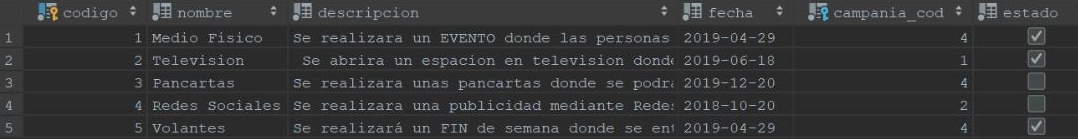
* **Llave Primaria:** La llave primaria es “código” el cual volverá único el identificador del evento.
* **Llaves Foráneas:** La llave foránea es “campania\_cod” la que llegará heredada de la tabla campaña, este es el código identificador de la campaña.

**3.4.3. DDL**

****

**Imagen5. DDL de la creación de la tabla EVENTO**

**3.4.4. Inserción de Datos**

****

**Imagen5.1. datos insertados en la tabla EVENTO**

**3.5. RECURSO**

Tabla en la cual se encuentra la descripción y características de los recursos con los cuales cuenta la empresa.

**3.5.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estará identificado cada uno de los recursos.
* **Nombre:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se especifica el nombre de cada uno de los recursos.
* **Tipo:** Es un atributo numérico tipo entero (INT), identificador del tipo de recurso.
* **Descripción:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se hace la respectiva descripción del nombre de cada recurso.
* **Valor:** Es un atributo obligatorio de tipo *DECIMAL* que hace referencia al precio promedio de los recursos.

**3.5.2. Llave Primaria y Llaves Foráneas**

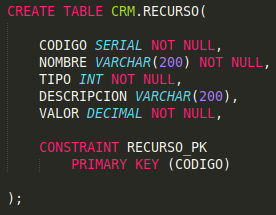
**Llave Primaria:**

* **RECURSO\_PK:** El código, es el identificador del recurso, usado para distinguir cada uno de ellos, y así tener pautas específicas del caso.

**Llaves Foráneas:**

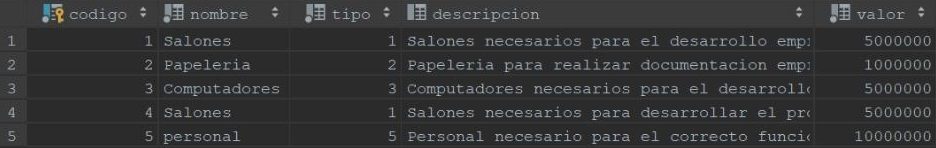
* La tabla RECURSO no recibe ningún dato de otra tabla, por el contrario, la tabla MEDIO es la que recibe como llave foránea la columna CÓDIGO.

**3.5.3. DDL**

****

**Imagen6. DDL de la creación de la tabla RECURSO**

**3.5.4. Inserción de Datos**

****

**Imagen6.1. datos insertados en la tabla RECURSO**

**3.6. MEDIO**

Tabla en la cual se encuentra la descripción y características de los medios publicitarios disponibles para las respectivas publicaciones de las campañas de la empresa.

**3.6.1. Atributos**

* **Código:** Es un atributo obligatorio numérico de tipo *SERIAL* e irrepetible con el que estará identificado cada uno de los medios.
* **Nombre:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se especifica el nombre de cada uno de los medios.
* **Descripción:** Es un atributo obligatorio de tipo *VARCHAR* de máximo 200 caracteres en donde se hace la respectiva descripción del nombre de cada uno de los medios.
* **Código Tipo de Medio:** Atributo heredado de la tabla TIPO\_MEDIO, especifica el código asignado a cada tipo de medio publicitario. Sus posibles estados se definen de la siguiente manera:
* **Código de Evento:** Atributo heredado de la tabla EVENTO, especifica el código asignado a cada uno de los eventos.
* **Código de Recurso:** Atributo heredado de la tabla RECURSO, especifica el código asignado a cada uno de los recursos.
* **Valor:** Es un atributo obligatorio de tipo *DECIMAL* que hace referencia al precio promedio de cada uno de los medios publicitarios.
* **Estado:** Es un atributo obligatorio de tipo *BOOLEAN* que hace referencia al estado de cada uno de los medios publicitarios. Sus posibles estados se definen de la siguiente manera:

**3.6.2. Llave Primaria y Llaves Foráneas**

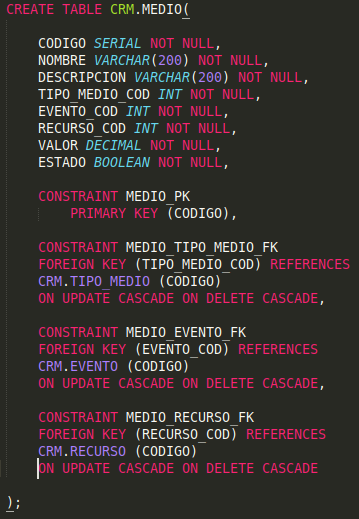
**Llave Primaria:**

* **MEDIO\_PK:** El CÓDIGO es el identificador del medio, por el cual se distingue cada uno de ellos con sus respectivos datos.

**Llaves Foráneas:**

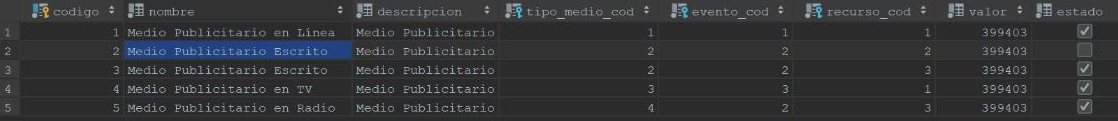
* **MEDIO\_TIPO\_MEDIO\_FK:** En la columna **TIPO\_MEDIO\_COD** almacenaremos los tipos de medios heredados de la tabla TIPO\_MEDIO, el cual hará más completa la información acerca del medio publicitario.
* **MEDIO\_EVENTO\_FK:** En la columna **EVENTO\_COD** almacenaremos los códigos de los eventos que se llevarán a cabo, de esta manera, al evento se le asignará un medio publicitario.
* **MEDIO\_RECURSO\_FK:** En la columna **RECURSO\_COD** almacenaremos los códigos de los recursos con los que cuenta la empresa para propagar la información.

**3.6.3. DDL**

****

**Imagen7. DDL de la creación de la tabla MEDIO**

**3.6.4. Inserción de Datos**

****

**Imagen7.1. datos insertados en la tabla MEDIO**

USTA Empresarial The Alirios

ESPECIFICACIÓN FUNCIONAL

CRM

Versión 1.1

**TABLA DE CONTENIDO EJB CRM FACHADA**

INTRODUCCIÓN

2. EJB

2.1. Tabla periodo

2.2. Tabla campania

2.3. Tabla evento

2.4. Tabla medio

2.5. Tabla recurso

2.6. Tabla tipo medio

**INTRODUCCIÓN**

Se crearon las secciones de fachada, been y servicio de CRM. Donde se crean los métodos del CRUD por cada una de las tablas de este parte de el proyecto.

1. **TABLA DE CONTENIDO EJB CRM FACHADA**
   1. **Tabla Periodo**
   2. **Tabla Campaña**

Se crean los siguiente métodos para la tabla recurso en cada una de las secciones:

**Fachada:** Interfaz utilizada para la declaracion de metodos.

* Listar Campaña:



* Crear Campaña:



* Editar Campaña:



* Eliminar Campaña:



* Buscar Campaña por nombre:



**Bean:** En el bean se hace la funcionalidad de los métodos declarados en la fachada.

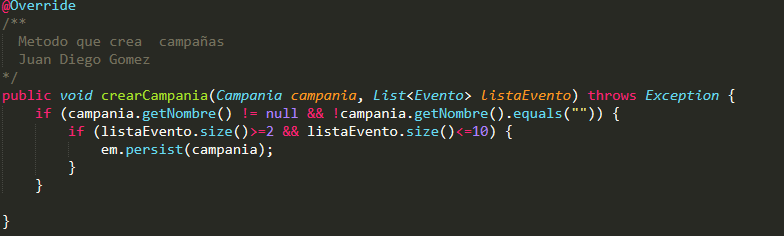
* Listar Campaña:

Método creado para listar todas las campañas publicitarias creadas este método contiene un Throws Exception lo que hace es que captura los posibles errores sobre la ejecución; además de eso utiliza una consulta hecha en el ORM pero se llama desde el modelo de CRM y retorna una lista tipo campania.



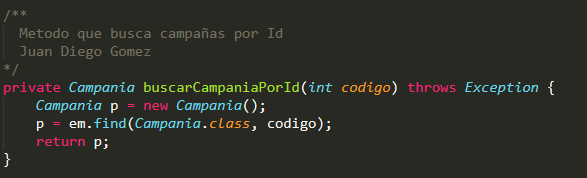
* Crear Campaña:

Método creado que nos permite crear una campaña, tiene como un parámetro un objeto de tipo Campania llamado campania y una lista de eventos, este método no retorna nada y tiene un Throws Exception que permite capturar errores posibles sobre la ejecución.



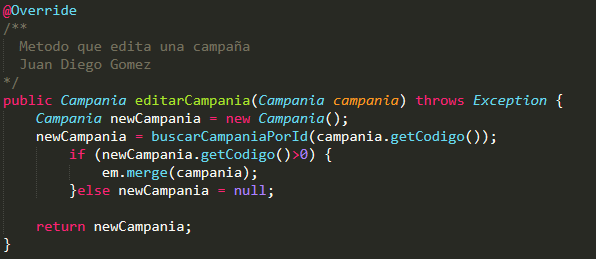
* Buscar Campaña por Id:

Es un método privado ya que solo lo usamos para retornar un objeto tipo campania que tenga el código que entra por parámetro.



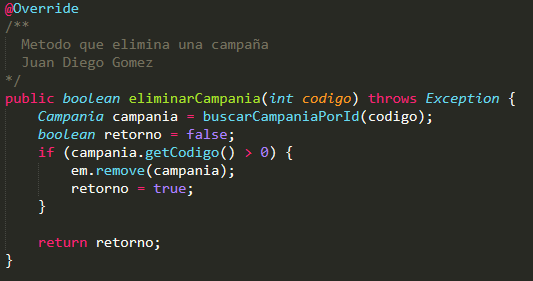
* Editar Campaña:

Este método permite editar una Campaña ya creada, para esto recibe como parámetro un objeto de tipo Campania, retornando un objeto tipo campania que se editado y actualizado. Tiene un Throws Exception que permite capturar errores posibles sobre la ejecución.



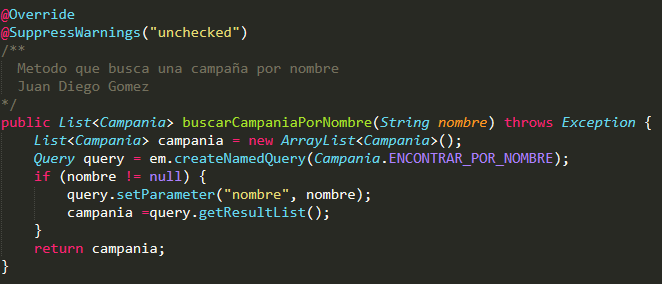
* Eliminar Campaña:

Este método elimina un objeto de tipo Campaña para esto tenemos un parámetro que es el código de la campaña que se quiere eliminar, el método es boolean ya que si lo borra retornara True y false si no es posible realizar la acción requerida y posee un Throws Exception que permite capturar errores posibles sobre la ejecución.



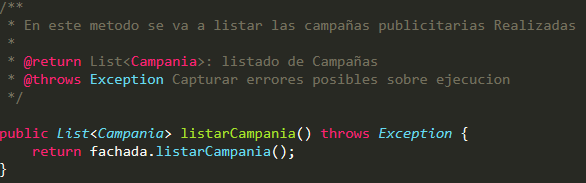
* Buscar Campaña por nombre:

Este método busca una Campaña por el nombre, tiene como parámetro un dato tipo String llamado nombre que será el nombre de la campaña y retorna una lista de campaña que contengan ese nombre y posee un Throws Exception que permite capturar errores posibles sobre la ejecución.

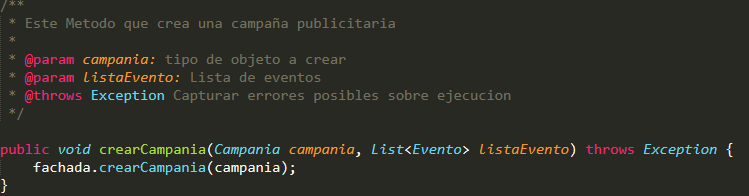


**Servicio:**

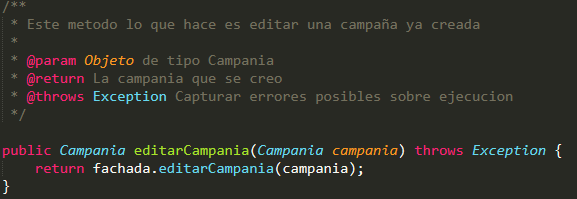
* Listar Campaña:



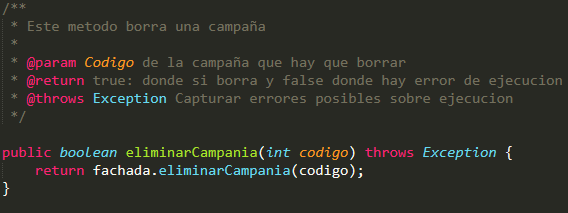
* Crear Campaña:



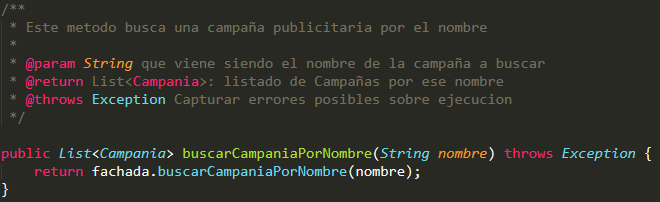
* Editar Campaña:



* Eliminar Campaña:



* Buscar Campaña por nombre:

****

**2.3 Tabla evento**

A Continuación se listan los métodos para cada una de las secciones

Fachada:

* public List<Evento> listarEvento() throws Exception;
* public void crearEvento(Evento evento) throws Exception;
* public Evento editarEvento(Evento evento) throws Exception;
* public boolean eliminarEvento(int codigo) throws Exception;
* public List<Evento> buscarEventoPorNombre(String nombre) throws Exception;

Servicio:

* public List<Evento> listarEvento() throws Exception {

return fachada.listarEvento();

}

* public void crearEvento(Evento evento) throws Exception {

fachada.crearEvento(evento);

}

* public Evento editarEvento(Evento evento) throws Exception {

return fachada.editarEvento(evento);

}

* public boolean eliminarEvento(int codigo) throws Exception {

return fachada.eliminarEvento(codigo);

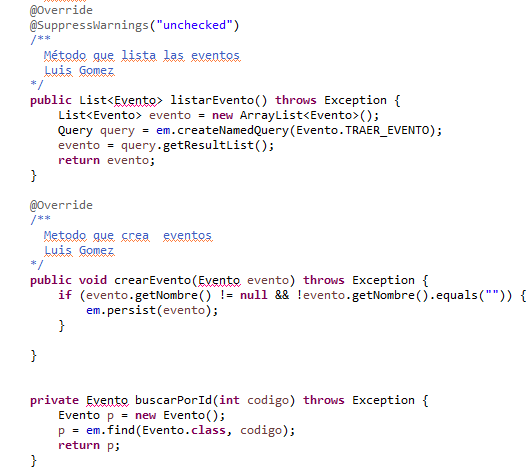
}

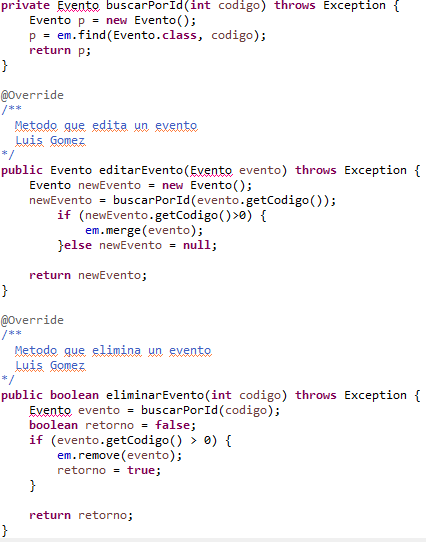
* public List<Evento> buscarEventoPorNombre(String nombre) throws Exception {

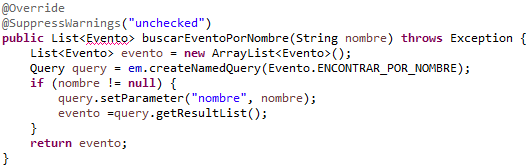
return fachada.buscarEventoPorNombre(nombre);

}

Bean:







Explicación de los métodos:

* listarEvento():

Este método lista los eventos publicitarios realizados. Retorna el listado de eventos y lanza una excepción para capturar errores posibles sobre ejecución.

* crearEvento(Evento evento):

Permite crear un nuevo evento publicitario. Recibe como parámetro el nuevo evento a crear y lanza una excepción para capturar errores posibles sobre ejecución.

* editarEvento(Evento evento):

Permite editar un evento ya creado. Recibe como parámetro el nuevo evento y retorna el evento que ha sido creado. Lanza una excepción para capturar errores posibles sobre ejecución.

* eliminarEvento(int codigo):

Permite eliminar un evento ya creado. Recibe como parámetro el código del evento a eliminar. Retorna **true** si el evento ha sido eliminado correctamente **false** en caso de error. Lanza una excepción para capturar errores posibles sobre ejecución.

* buscarEventoPorNombre(String nombre):

Permite buscar un evento. Recibe como parámetro el nombre del evento a buscar. Retorna una lista con los eventos que coinciden con la búsqueda. Lanza una excepción para capturar errores posibles sobre ejecución.

**2.4 Tabla medio**

Se crean los siguiente métodos para la tabla medio en cada una de las secciones:

Fachada

* public List<Medio> listarMedio() throws Exception
* public void crearMedio(Medio medio) throws Exception
* public Medio editarMedio(Medio medio) throws Exception
* public boolean eliminarMedio(int codigo) throws Exception
* public List<Medio> buscarMedioPorNombre(String nombre) throws Exception

Servicio

* public List<Medio> listarMedio() throws Exception {

return fachada.listarMedio();

}

* public void crearMedio(Medio medio) throws Exception {

fachada.crearMedio(medio);

}

* public Medio editarMedio(Medio medio) throws Exception {

return fachada.editarMedio(medio);

}

* public boolean eliminarMedio(int codigo) throws Exception {

return fachada.eliminarMedio(codigo);

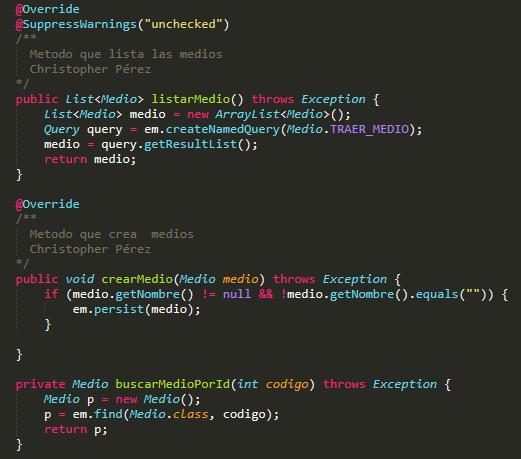
}

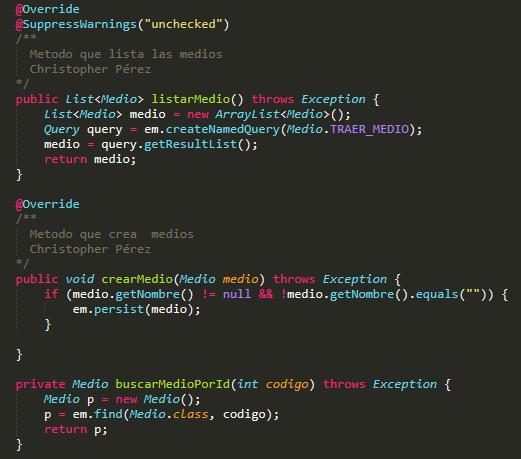
* public List<Medio> buscarMedioPorNombre(String nombre) throws Exception {

return fachada.buscarMedioPorNombre(nombre);

}

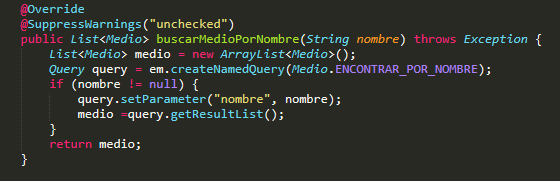
Bean











Explicaciòn de los métodos:

* listarMedio()

En este método permite listar los medios publicitarios utilizados, retorna listado de todos los medios sin pedir ningun parametro y tiene una Excepción que permite capturar errores posibles sobre ejecución

* crearMedio(Medio medio)

Este Método crea un objeto de tipo medio, pide un atributo tipo Medio llamado medio, no retorna nada pero tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* editarMedio(Medio medio)

Este método permite editar un medio ya creado, para esto recibe como parámetro un objeto de tipo Medio, retornando el medio que se editado y actualizado. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* eliminarMedio(int codigo)

Este método elimina un objeto de tipo Medio para esto pide como parámetro un dato int llamado codigo que representa o equivale al código del medio a eliminar. Si lo borra retornara True y false si no es posible realizar la acción requerida. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* buscarMedioPorNombre(String nombre)

Este método busca un medio publicitario por el nombre, para esto pide un String que viene siendo el nombre del medio a buscar y retorna una List<Medio> que es el listado de medios por ese nombre.Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

**2.5 Tabla recurso**

Se crean los siguiente métodos para la tabla recurso en cada una de las secciones:

Fachada

* public List<Recurso> listarRecursos() throws Exception;
* public void crearRecurso(Recurso puc) throws Exception;
* public Recurso editarRecurso(Recurso puc) throws Exception;
* public boolean eliminarRecurso(int codigo) throws Exception;
* public Recurso listarRecursoNombre(String nombre) throws Exception;

Servicio

* public List<Recurso> listarRecursos() throws Exception {

return fachada.listarRecursos();

}

* public void crearRecurso(Recurso puc) throws Exception {

fachada.crearRecurso(puc);

}

* public Recurso editarRecurso(Recurso puc) throws Exception {

return fachada.editarRecurso(puc);

}

* public boolean eliminarRecurso(int codigo) throws Exception {

return fachada.eliminarRecurso(codigo);

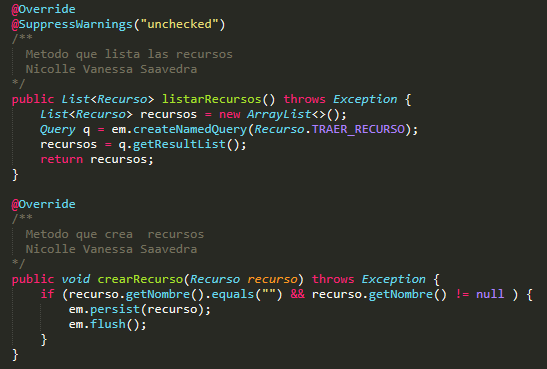
}

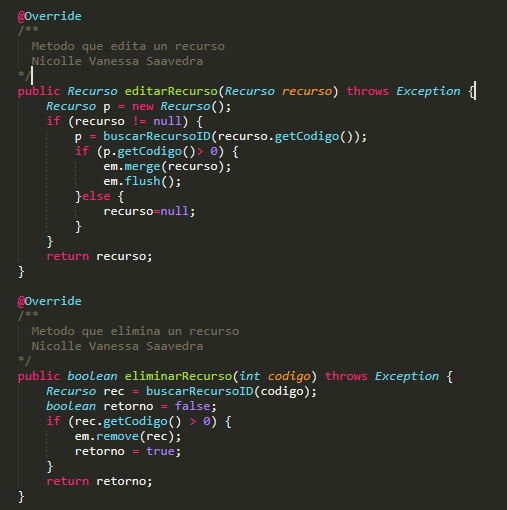
* public Recurso listarRecursoNombre(String nombre) throws Exception {

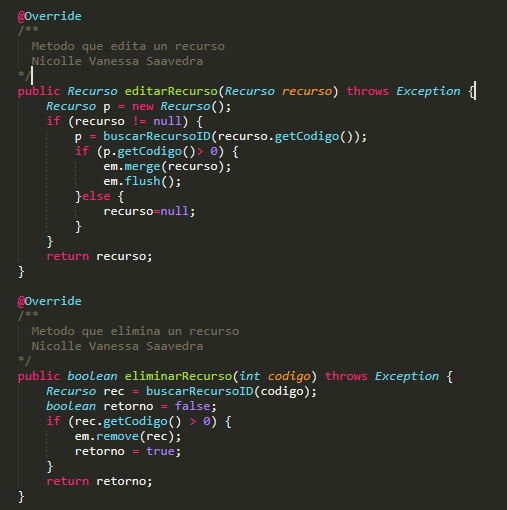
return fachada.listarRecursoNombre(nombre);

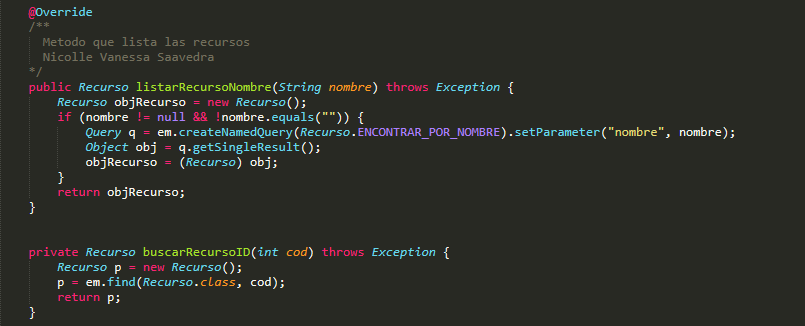
}

Bean









Explicaciòn de los métodos:

* listarRecursos()

Método creado para listar todos los recursos contenidos en la tabla. Contiene una Excepción Captura los posibles errores sobre la ejecución y retorna.

* crearRecurso(Recurso puc)

Método que permite crear un recurso, este método recibe como parámetro un objeto de tipo Recurso llamado puc, este método no retorna nada. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* editarRecurso(Recurso puc)

Método que permite actualizar un recurso, recibe como parámetro un objeto de tipo recurso y retorna el mismo objeto pero con las modificaciònes realizadas. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* eliminarRecurso(int codigo)

Método que permite eliminar un recurso, recibe como parámetro un entero que representa el código del recurso y retorna true: Si se borró y false si no se borró. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

* listarRecursoNombre(String nombre)

Método que busca una recurso por nombre, recibe como parámetro un dato de tipo string (nombre) que equivale a el nombre del recurso que va buscar y retorna objeto de tipo recurso. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.

**2.6 Tabla Tipo Medio**

Se crean los siguiente métodos para la tabla Tipo Medio en cada una de las secciones:

**Fachada:**

* public List<TipoMedio> listarTipoMedio() throws Exception;
* public void crearTipoMedio(TipoMedio tipo) throws Exception;
* public TipoMedio editarTipoMedio(TipoMedio tipo) throws Exception;
* public boolean eliminarTipoMedio(int codigo) throws Exception;
* public TipoMedio listarTipoMedioNombre(String nombre) throws Exception;

**Servicio**

* public List<TipoMedio> listarTipoMedio() throws Exception {

return fachada.listarTipoMedio();

}

* public void crearTipoMedio(TipoMedio tipo) throws Exception {

fachada.crearTipoMedio(tipo);

}

* public TipoMedio editarTipoMedio(TipoMedio tipo) throws Exception {

return fachada.editarTipoMedio(tipo);

}

* public boolean eliminarTipoMedio(int codigo) throws Exception {

return fachada.eliminarTipoMedio(codigo);

}

* public TipoMedio listarTipoMedioPorNombre(String nombre) throws Exception {

return fachada.listarTipoMedioPorNombre(nombre);

}

**Bean:** En el bean se hace la funcionalidad de los métodos declarados en la fachada.

Listar Tipo Medio:

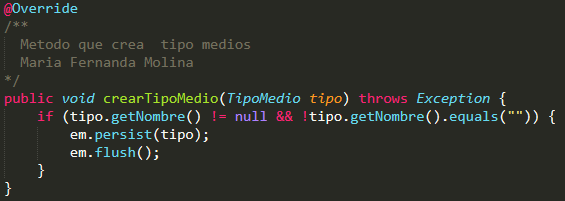
* listarTipoMedio()

Método creado para listar todos los Tipo medio contenidos en la tabla. Contiene una Excepción Captura los posibles errores sobre la ejecución y retorna.



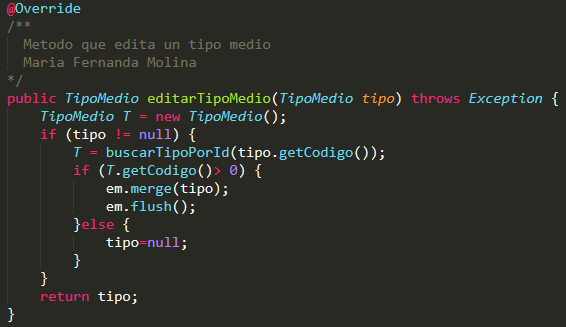
* crearTipoMedio(TipoMedio tipo)

Método que permite crear un Tipo Medio, este método recibe como parámetro un objeto de tipo TipoMedio llamado tipo, este método no retorna nada. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.



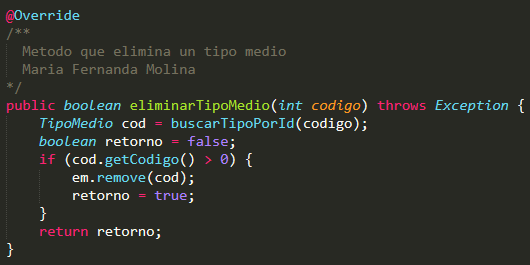
* editarTipoMedio(TipoMedio tipo)

Este método permite editar un Tipo Medio ya creado, para esto recibe como parámetro un objeto de tipo Tipo Medio, retornando el tipo que se editado y actualizado. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.



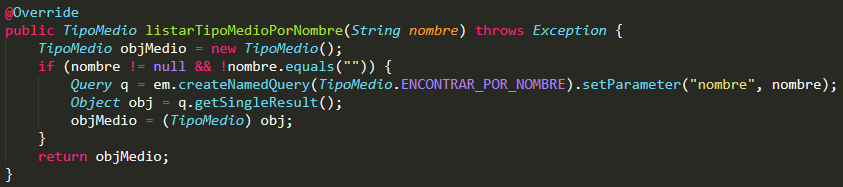
* eliminarTipoMedio(int codigo)

Método que permite eliminar un Tipo Medio, recibe como parámetro un entero que representa el código del Tipo Medio y retorna true: Si se borró y false si no se borró. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.



* listarTipoMedioPorNombre(String nombre)

Método que busca una Tipo Medio por nombre, recibe como parámetro un dato de tipo string (nombre) que equivale a el nombre del Tipo Medio que va buscar y retorna objeto de tipo Tipo Medio. Tiene una Excepción que permite capturar errores posibles sobre la ejecución.



**2.6 Tabla periodo**

**Fachada:**

* public List<Periodo> listarPeriodo() throws Exception;
* public void crearPeriodo(Periodo per) throws Exception;
* public Periodo editarPeriodo(Periodo per) throws Exception;
* public boolean eliminarRegistros(int codigo) throws Exception;

**Servicio**

public List<Periodo> listarPeriodo() throws Exception {

return fachada.listarPeriodo();

}

public void crearPeriodo(Periodo per) throws Exception {

fachada.crearPeriodo(per);

}

public Periodo editarPeriodo(Periodo per) throws Exception {

return fachada.editarPeriodo(per);

}

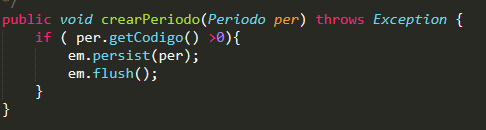
public boolean eliminarRegistros(int codigo) throws Exception {

return fachada.eliminarRegistros(codigo);

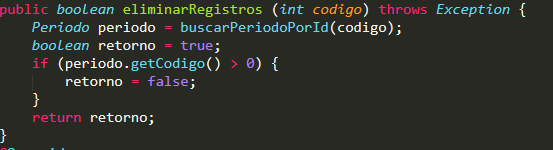
}

**Bean:** En el bean se hace la funcionalidad de los métodos declarados en la fachada.

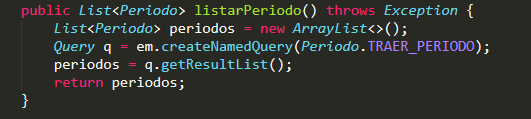
* crear período: método que permite crear un periodo este método recibe un parámetro de tipo periodo y no retorna nada.



* eliminar registros: este método permite eliminar los registros que tenga la tabla por el código revise un parámetro entero utiliza el método buscar por id y si encuentra ese codigo lo convierte en falso o inactivo



* listar periodo: este método lista todo lo que tenga la tabla hay un arraylist de tipo periodos hace la consulta de traer periodo y retornar el arraylist de tipo periodo



* buscar periodo por id: este método busca el periodo por id recibe un parámetro de tipo entero que corresponde al código que desea buscar, retorna el objeto de tipo periodo

