

# DBCASE WEB 3.0

*“Una herramienta para el diseño de Bases de Datos.”*

*Trabajo de fin de grado del  
Grado de Ingeniería del Software  
Curso Académico 2020 / 2021*



*Facultad de Informática  
Universidad Complutense de Madrid*

*Autora Roxanne A. Caiafa*

*Profesor director Fernando Sáenz Pérez*

## **Agradecimientos.**

A Fernando Sáenz Pérez director de proyecto, por su dedicación y confianza desde el inicio del desarrollo de este proyecto, que en todo momento estuvo disponible para ayudarme y aconsejarme en todo lo relacionado a esta aplicación, incluso con todas las dificultades de distancia producidas por la pandemia en este curso académico.

A todos los profesores con los que he contado a lo largo de estos años por su guía y retos, por su constancia y paciencia. Gracias por enseñarme a pensar, gracias por darme las herramientas para saber cómo resolver y/o abordar casi cualquier tipo de problema, por hacerme resiliente, independiente y en algunos casos autodidacta.

A mi madre y a mis compañeros, quienes se han convertido en amigos para toda la vida y que tanto me ayudaron, apoyaron, confiaron en mi, por animarme en los momentos más difíciles, por ser familia y por tanto cariño Sergio, Eloy, Miguel, Victoria, Víctor, Leila, Sofía, Aurelio.

Sin la existencia de todas estas bellísimas personas en mi camino no hubiese sido posible haber llegado hasta aquí.

# Índice

Resumen .....	5
Abstract.....	6
1 Introducción.....	7
1.1 Antecedentes.....	8
1.2 Objetivos.....	9
1.3 Plan de trabajo.....	10
2 Tecnologías .....	11
2.1 Frameworks .....	11
2.2 Lenguajes .....	12
2.1.1. HTML5 (HyperText Markup Language).....	12
2.2.1 JavaScript .....	12
2.2.2 Java.....	13
2.3 Librerías.....	13
2.3.1 Vis Js .....	13
2.3.2 JQuery .....	13
2.3.3 JsPDF .....	14
2.4 Motor de plantillas.....	14
2.4.1 Thymeleaf .....	14
3 Diseño .....	15
3.1 Arquitectura.....	15
3.2 Diagramas de clases.....	16
3.2.1 Modelo.....	16
3.2.2 Diagramas de clase paquete Controlador.....	23
4 Desarrollo.....	26
4.1 Mejoras visuales.....	26
4.2 Configuración.....	29
4.3 Solución de problemas con la traducción de esquemas lógico y físico .....	30
4.4 Nuevas funcionalidades .....	31
4.5 Ejemplos de traducción.....	37
4.5.1 Atributos multivalorados .....	37
4.5.2 Participación total.....	38
4.5.3 Entidad débil .....	40
4.5.4 Relación recursiva .....	41
4.5.5 Relación isA .....	43
5 Conclusiones y trabajo futuro.....	45

5.1	Trabajo futuro .....	46
5.1.1	Funcionalidades futuras.....	46
6	Repositorio.....	46
7	Bibliografía .....	47
8	Apéndice - Manual de usuario .....	48
8.1	Inicio de sesión.....	48
8.2	Menú.....	49
8.2.1	Ayuda .....	49
8.2.2	Diseño .....	50
8.2.3	Opciones .....	51
8.2.4	Archivo .....	52
8.2.5	Dibujar.....	55
8.3	Funcionalidad sobre el esquema conceptual.....	59
8.3.1	Entidad .....	59
8.3.2	Relación.....	63
8.3.3	IsA.....	66
8.3.4	Atributo.....	67
8.4	Esquema lógico .....	68
8.4.1	Generar .....	68
8.4.2	Guardar como .....	69
8.5	Esquema físico .....	69
8.5.1	Generar .....	70
8.5.2	Guardar como .....	70
8.6	Introducción al sistema.....	71

# Resumen

DBCASE Web 3.0 es una nueva versión del proyecto DBCASE Web cuyo objetivo es diseñar diagramas Entidad-Relación, de manera intuitiva, al poder construir bases de datos relacionales, permitiendo comprobar posteriormente su traducción y además generar el código asociado a las tablas creadas.

Esta aplicación web permite, a los usuarios con cuenta UCM, disfrutar de una herramienta que da acceso al diseño de base de datos de una manera práctica incluso para los que realizan su primer diseño. Así que para los que comienzan con la teoría de bases de datos relacionales, esta herramienta marca la diferencia, porque después de leer la teoría y tener nociones básicas, la experiencia de poder crear diagramas Entidad-Relación de la forma que ofrece DBCase Web, optimizando el tiempo, el material y la energía, hace que diseñar y aprender a diseñar base de datos relacionales se haga ligero. Si además tenemos en cuenta que ese diagrama Entidad-Relación se traduce a esquema lógico y a esquema físico, hace de este proyecto la herramienta clave para crear, afianzar y ratificar conocimientos.

La finalidad principal de este proyecto es mejorar e implementar nuevas funcionalidades con respecto a la versión anterior.

## **Palabras Clave**

Bases de datos | Java | JavaScript | Modelo Entidad Relación | Modelo Relacional |  
Aplicación web | SQL | Spring Boot | Bootstrap | JQuery

# Abstract

DBCCase Web 3.0 is a new versión of the DBCase Web Project, whose objective is to design entity-relation diagrams, in a intuitive way and to be able to build relational databases, later check their translation and also generate the code associated with the tables created.

This web app allows users with UCM account to enjoy a tool that gives access to database design in a very intuitive way, even for those who make their first design. So for those who start with relation database theory, this tool makes the difference, because after reading theory and having basic notions, the experience of being able to create Entity-Relation diagrams in the way that DBCase Web offers, optimizing the time, material and energy, makes designing and learning to design relational databases in a light way. If we also take into account that this Entity-Relation diagram translates into a logical scheme and physical scheme, it makes this Project the key tool to create, consolidate and ratify knowledge.

The main purpose of this Project is to improve and implement new functionalities with respect to the previous versión.

## **Keywords**

Bases de datos | Java | JavaScript | Modelo Entidad Relación | Modelo Relacional | Aplicación web | SQL | Spring Boot | Bootstrap | JQuery

# 1 Introducción

DBCASE Web 3.0 continúa la implementación del proyecto DBCASE cuyo desarrollo empezó en el año 2007. En esta versión se implementan nuevas funcionalidades, corrección de errores y nuevos cambios en su interfaz de usuario con el objetivo de facilitar el uso y mejorar la experiencia de usuario.

Este proyecto ha sido desarrollado para poder dar una herramienta, a los alumnos UCM, que permita entender mejor el funcionamiento de las bases de datos relacionales, teniendo claro los principios de los diagramas Entidad-Relación, es muy intuitivo diseñar. Es un medio útil tanto para los que comienzan en el mundo del diseño de bases de datos, como para diseñadores expertos de manera que construir una base de datos relacional se pueda realizar de forma fácil y rápida.

## 1.1 Antecedentes

La aplicación DBCASE Web 3.0 precede a varias versiones realizadas por compañeros en años pasados.

Se detalla a continuación.

(2007 - 2008) DBDT de sistemas informáticos

Autores:

- Alberto Milán Gutierrez Miguel Martinez Segura
- Francisco Javier Cáceres González
- Yolanda García Ruiz (Tutor)

(2008 - 2009) BDCASE de sistemas informáticos

Autores:

- Rodrigo Denís Cepeda Mateos
- Cristina Marco De Francisco
- Tello Serrano Gordillo
- Fernando Sáenz Pérez (Tutor)

(2018-2019) DBCASE 2.0 (2018/2019) Autores:

- Miguel Arriba García
- Fernando Sáenz Pérez (Tutor)

(2019-2020) DBCASE Web 2.0

Autores

- Luis Eduardo Paucar Cerrón
- Yrving David Conde Cubas
- Fernando Sáenz Pérez (Tutor)

La versión, realizada en el año anterior contaba con el objetivo de convertir la aplicación de escritorio en una aplicación web, para así ser compatible con cualquier dispositivo y cuyo único requisito para su acceso, sea un navegador web, evadiendo así problemas de versiones no compatibles o dependencia de software específico para su ejecución.

## 1.2 Objetivos

DBCASE (Database Computer-Aided Software Engineering, [dbcase.sourceforge.net](http://dbcase.sourceforge.net)) ofrece ser una herramienta para el diseño de bases de datos relacionales que integra las tres etapas clásicas de diseño (conceptual, lógico y físico), dando soporte a distintos gestores (Oracle, MySQL, Access). Permite el diseño gráfico de diagramas entidad-relación y traducciones automáticas a los esquemas lógico (modelo relacional) y físico (SQL). Comenzó siendo una aplicación de escritorio implementada en Java y multiplataforma, ahora es una aplicación web implementada en Java y JavaScript principalmente. El objetivo del proyecto actual es ampliar sus funcionalidades para:

- 1) Incluir nuevas construcciones de diseño conceptual (agregaciones, distintas notaciones para las cardinalidades).
- 2) Generación de esquemas lógico y físico con estas nuevas construcciones.
- 3) Mejoras visuales como el contraste de colores en el modo oscuro.
- 4) Permitir al usuario ver el nombre del archivo con el que se está trabajando o “nuevo documento” en el caso de ser un nuevo documento.
- 5) Mejoras en el comportamiento como:
  - Limpiar paneles del esquema lógico y físico si se quiere trabajar en un nuevo diseño.
  - Corregir la traducción de las entidades débiles.
  - Configuración para ejecutarse con cualquier cuenta ucm.es
  - Mostrar las formas seleccionadas con mejor contraste.
  - Al editar la cardinalidad, actualizar en el cuadro de diálogo, que aparezcan los valores de Cardinalidad y Participación mínimo-máximo que tuviese en el diagrama
  - Solucionar problemas de traducción en las restricciones de integridad referencial
- 6) Implementación de participación total o parcial en la pantalla de cardinalidad y participación.

## 1.3 Plan de trabajo

Fases que han tenido lugar durante todo el proyecto.

### 1. Toma de contacto con la aplicación:

Esta primera fase se ha comprobado el estado de la aplicación. Para comenzar a estudiar las funcionalidades, antes ha tenido lugar el estudio de la teoría de las bases de datos relacionales, más concretamente, el diseño lógico y el modelo relacional. Teoría ha sido proporcionada por Fernando Sáenz Pérez.

Y para terminar esta fase, se han generado diseños de ejemplo de todo tipo, agregación, cardinalidades 1-1,1-N,N-M, participación total o parcial, entidades débiles, relación IsA. Todas estas pruebas se añadieron a una carpeta que se ha ido actualizando a lo largo del proyecto. Por otra parte, esta fase ha finalizado con un fichero que contenía todas las anotaciones de posibles mejoras y corrección de errores.

### 2. Desarrollo de mejoras y corrección de errores:

Esta segunda fase, comenzó una vez que la fase de toma de contacto fue suficientemente avanzada y por tanto finalizada.

Para comenzar con el desarrollo lo primero que hice fue intentar compilar el código y al ver que no lograba compilarlo, contacté con uno de los participantes del proyecto previo, quien gustosamente me ayudó a configurar el IDE Eclipse, y a explicarme la organización del código. También me ayudó a crear las credenciales como administradora de la API que se encarga del inicio de sesión con la cuenta UCM. A partir de tener un IDE configurado, se pudo proceder al desarrollo del código, cumpliendo la lista de objetivos.

### 3. Pruebas:

La fase de pruebas iba en paralelo con la fase desarrollo, de modo que en determinados periodos de tiempo se realizaba una entrega de uno o varios puntos de la lista creada en la fase uno. Además, esta lista a veces crecía a medida que se hacían pruebas.

## 2 Tecnologías

El objetivo principal de este proyecto es mejorar la apariencia y funcionalidad de la web, para ello hay que hablar del framework y las librerías específicas que han permitido realizar distintas acciones como la creación de los elementos del modelo entidad relación, generar ficheros PDF, los menús emergentes, interfaz de usuario, entre otros.

### 2.1 Frameworks

#### **Spring boot**

En su transición de aplicación de escritorio a aplicación web, se optó por crear el proyecto con la herramienta Spring boot. Para hablar de esta herramienta, hay que hacer una pequeña introducción de Spring.

Spring es un framework Open Source que facilita la creación de aplicaciones de todo tipo en Java, Kotlin y Groovy. Este framework está dividido en diversos módulos que podemos utilizar ofreciéndonos muchas más funcionalidades, esto le hace caracterizarse por tener una estructura modular y una gran flexibilidad para implementar diferentes tipos de arquitectura según las necesidades de la aplicación, sin embargo, hay que hacer una serie de configuraciones necesarias para el uso de estos módulos que se ofrecen.

Spring boot nace con la intención de simplificar esta serie de configuraciones, en concreto las configuraciones de archivos XML, de esta forma para crear un proyecto con Spring, utilizando Spring boot, se puede centrar desde un principio en el desarrollo o codificación de la aplicación y no dedicar tanto tiempo a las configuraciones o tareas de deployment, ya que además ayuda a administrar todas las dependencias (archivos JAR y versiones compatibles).

#### **Bootstrap**

Es un framework Open Source de diseño web que facilita la utilización de componentes HTML, CSS y JavaScript ya creados y se pueden extender. Con este framework se pueden hacer diseños web adaptables, es decir, un diseño adaptable a las diferentes pantallas de dispositivos que utilice el usuario.

## 2.2 Lenguajes

### 2.1.1. HTML5 (HyperText Markup Language)

HTML5 es un lenguaje de marcado estándar, la última versión hasta el momento de HTML. Una de las ventajas que ofrece esta versión es que incluye una API de JavaScript para mejorar el almacenamiento, contenido multimedia y el acceso al hardware.

Además, este lenguaje permite añadir scripts, que ofrecen instrucciones específicas a los navegadores que se encargan de procesar el lenguaje, y así hacerlos más ligeros. Entre los scripts que pueden generarse, los más conocidos y utilizados son JavaScript y PHP, en este proyecto se ha utilizado JavaScript, y será descrito más adelante.

Para modificar la apariencia de HTML, se ha utilizado CSS3 (Cascade Style Sheet) o también conocido como Hoja de estilos en cascada, que permite cambiar el aspecto de documentos estructurado escrito en un lenguaje de marcado como lo es HTML, permitiendo así, diferenciar entre estilos y estructura. A través de esta característica, se permite que simplemente cambiando las reglas CSS, se alternar totalmente el aspecto de un sitio web, siendo posible, además, que los desarrolladores puedan hacer uso de hojas de estilo personalizadas.

CSS3, la versión más reciente de CSS, se publicó en el año 2011 y, hasta día de hoy, contiene las especificaciones más completas y amplias. El aumento de funcionalidades que presenta ha hecho que la versión 1 y 2 hayan sido eclipsadas trayendo, además, la división del código en módulos.

### 2.2.1 JavaScript

JavaScript es el lenguaje de programación encargado de dar de mayor interactividad y eficacia a las páginas web. Cuando JavaScript se ejecuta en el navegador, no precisa de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le revela como uno de los tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura).

JavaScript no solo se utiliza en el lado del cliente, sino que junto a una tecnología llamada AJAX permite intercambiar información con el servidor sin tener que recargar la página. Es decir, sólo cargamos de la página lo necesario. Esta tecnología desarrollada en JavaScript ha supuesto uno de los primordiales avances en el desarrollo web.

### 2.2.2 Java

Es un lenguaje de propósito general y orientado a objetos, además tiene un fuerte tipado. Una vez compilado el programa, este se convierte a bytecode de Java y este bytecode es leído por el JVM (Java Virtual Machine), que es el que se encarga de traducirlo a código máquina.

Java es un lenguaje de programación y una plataforma informática, distribuida por primera vez en 1995 por Sun Microsystems. Hay numerosas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde teléfonos móviles hasta Internet, desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, Java está en todas partes.

## 2.3 Librerías

Para esta aplicación se utilizan varias librerías que facilitan tanto el desarrollo gráfico como de las herramientas para poder diseñar un modelo entidad-relación con un amplio abanico de opciones, también otras funcionalidades.

### 2.3.1 Vis Js

Es una librería Javascript que permite crear redes con nodos y aristas. Admite añadir distintas formas de nodos, estilos, colores, tamaños, imágenes y formas personalizadas. La visualización de la red sirve en cualquier navegador ya que utiliza un transcompilador llamado Babel que permite convertir el código Javascript en una versión, del mismo, compatible con versiones anteriores que pueden ejecutar los motores JavaScript más antiguos. Vis JS tiene doble licencia en Apache 2.0 License y The MIT License.

Esta librería ha permitido realizar mucho de los gráficos aunque no la mayoría, por lo que se han añadido funcionalidades como el subrayado, movimiento de elementos, dobles líneas, entre otros, a las figuras de los nodos.

### 2.3.2 JQuery

Es una librería Javascript que nace de la necesidad de tener una colección de recursos que permite implementar funcionalidades o efectos sin la necesidad de escribir todo el código que generalmente sería necesario. Esta librería permite manipular documentos HTML, manejar eventos, animación y Ajax entre otros. Teniendo compatibilidad con una amplia variedad de navegadores, su constante crecimiento y su fácil escritura y comprensión del código, siendo Open Source, se convierte en una librería que conviene mucho utilizar.

### 2.3.3 JsPDF

Es una librería Javascript que permite generar archivos PDF a partir de código HTML.

## 2.4 Motor de plantillas

### 2.4.1 Thymeleaf

Es un motor de plantillas, es decir es una tecnología que permite definir una plantilla y junto a un modelo de datos tener un nuevo documento, sobre todo en entornos web. Está basado en añadir atributos o etiquetas HTML, por lo que permite que las plantillas se rendericen en local y esa misma plantilla utilizarla para que sea procesada dentro del motor de plantilla, esto facilita que las tareas de diseño y de programación puedan llevarse conjuntamente.

Es integrable con varios frameworks y en este proyecto se ha utilizado para facilitar el cambio de idioma y la visualización de temas (claro u oscuro), según desee el usuario.

## 3 Diseño

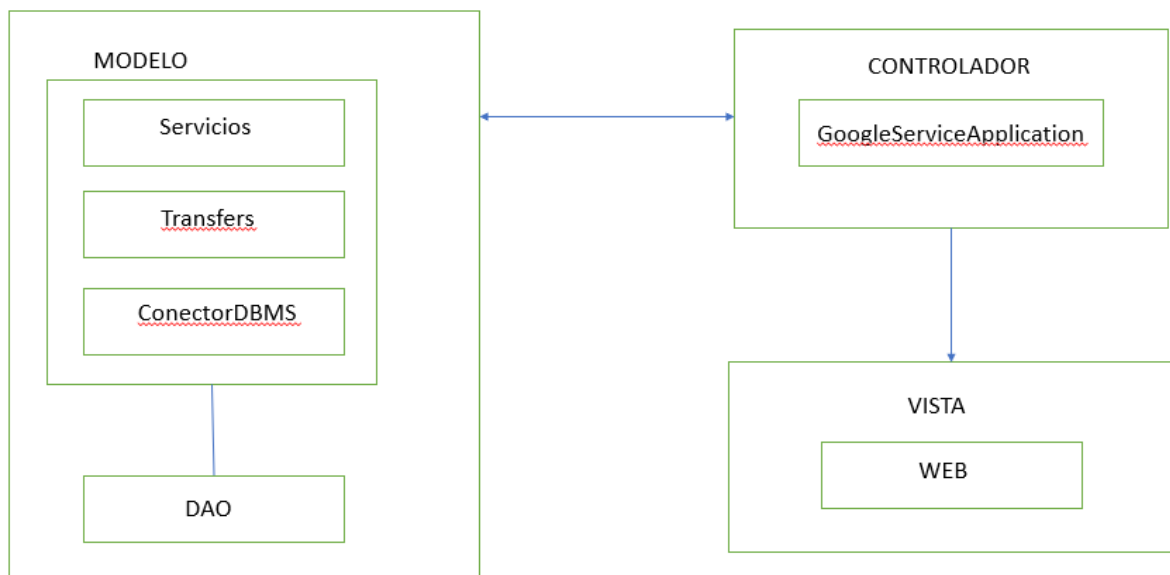
### 3.1 Arquitectura

Para entender mejor dónde se han hecho cambios, este apartado hace una breve introducción de la arquitectura del sistema.

La aplicación tiene una estructura multicapa cuyas ventajas son la modularidad y adaptabilidad. Se ve reflejado al haberse pasado de una aplicación de escritorio a una aplicación web, donde se dejó la lógica principal y se adaptó una nueva vista.

Los patrones de diseño utilizados para la implementación de esta arquitectura son:

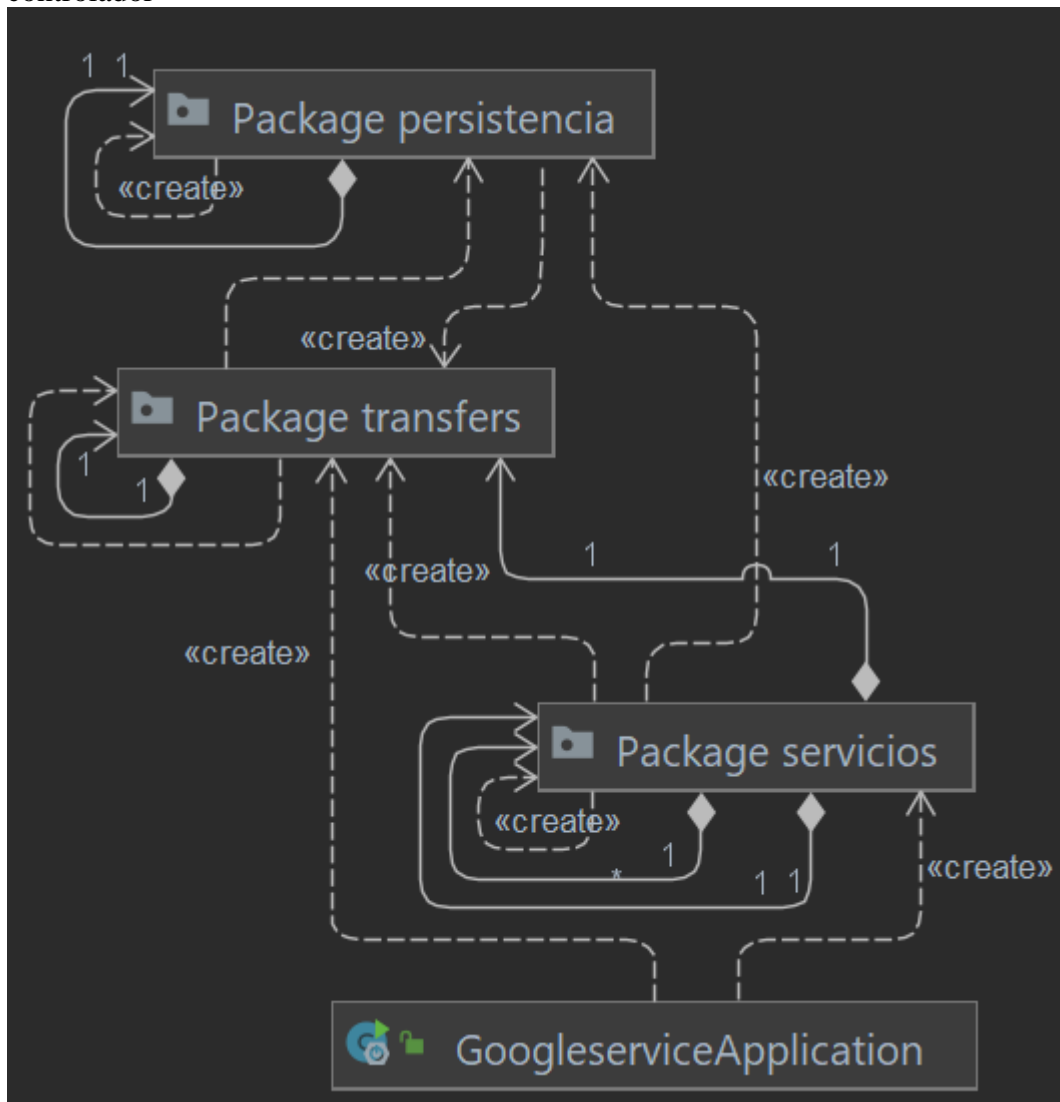
- Modelo vista controlador (MVC).
- Patrón transferencia (Transfer).
- Patrón objeto de acceso a datos (DAO).



Se desarrolla una comunicación entre el Modelo y el Controlador. La Vista recibe datos a mostrar del Controlador.

## 3.2 Diagramas de clases

El siguiente diagrama es una vista de cómo los paquetes del modelo se relacionan con el controlador



### 3.2.1 Modelo

El modelo consta de cuatro paquetes :

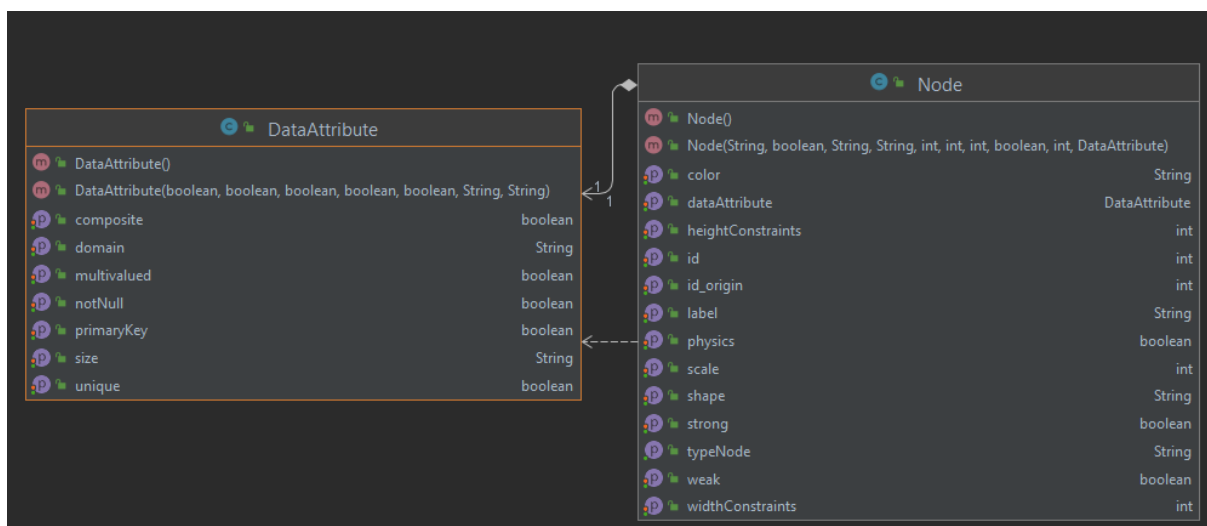
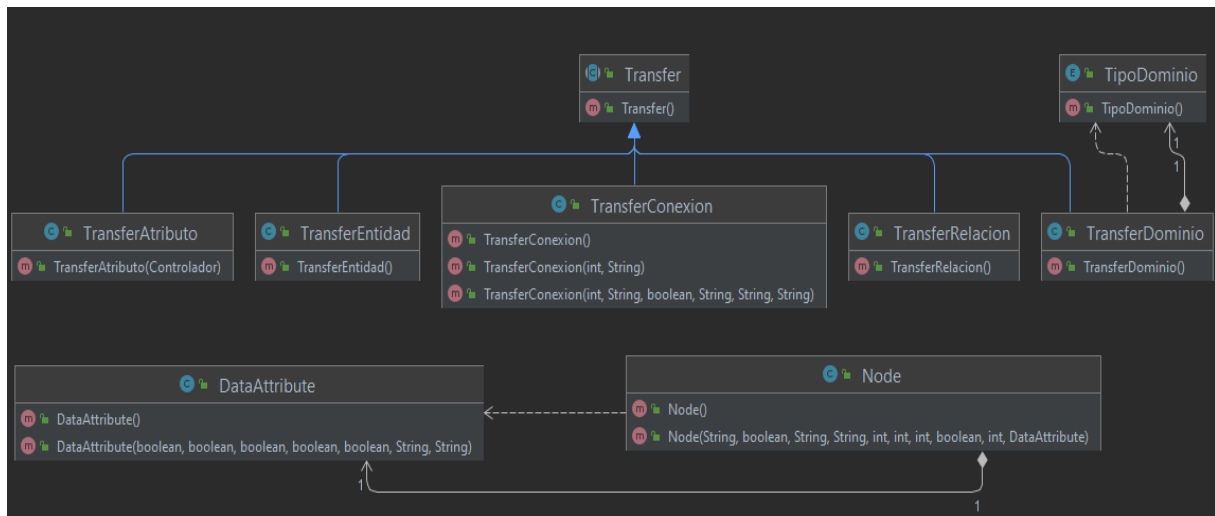
- Persistencia
- Transfers
- ConectorDBMS
- Servicios

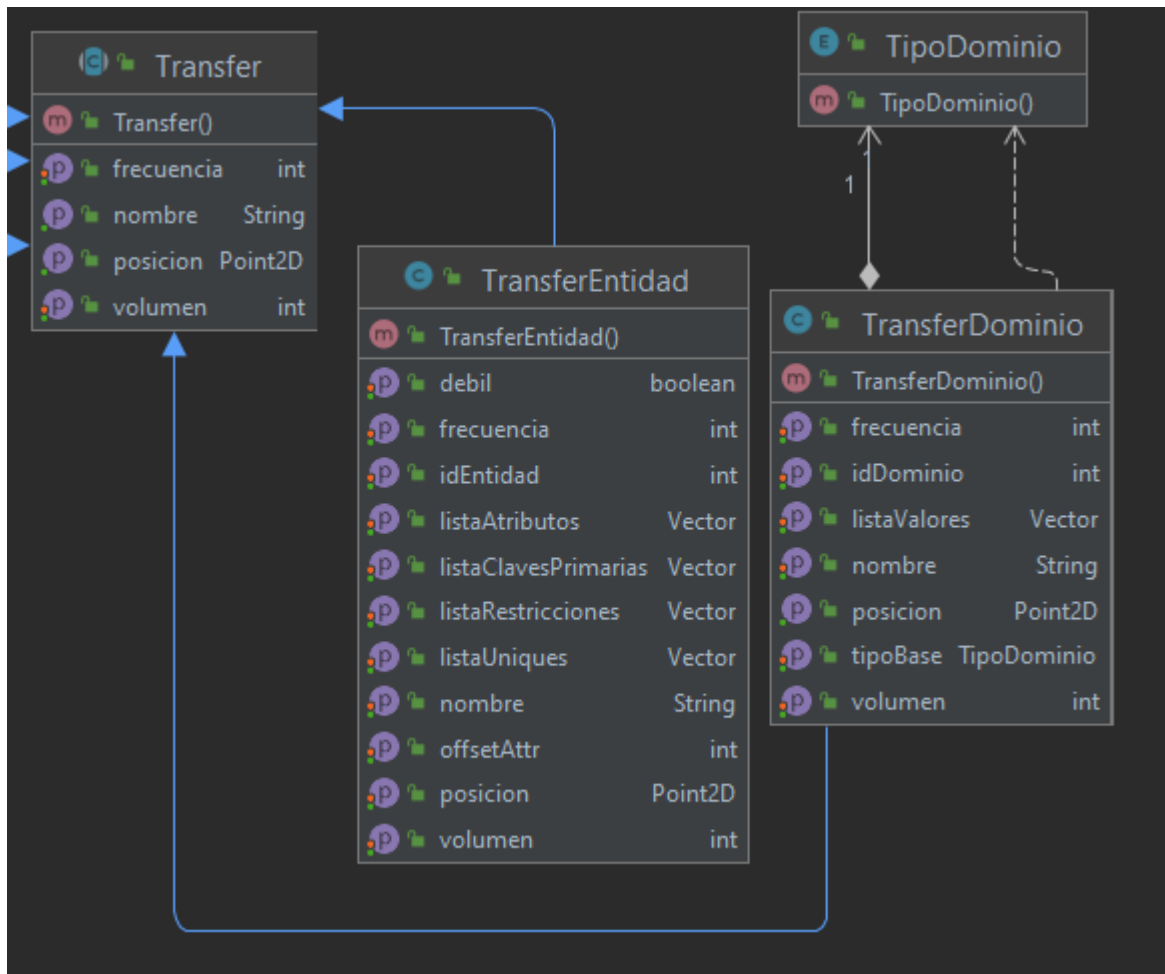
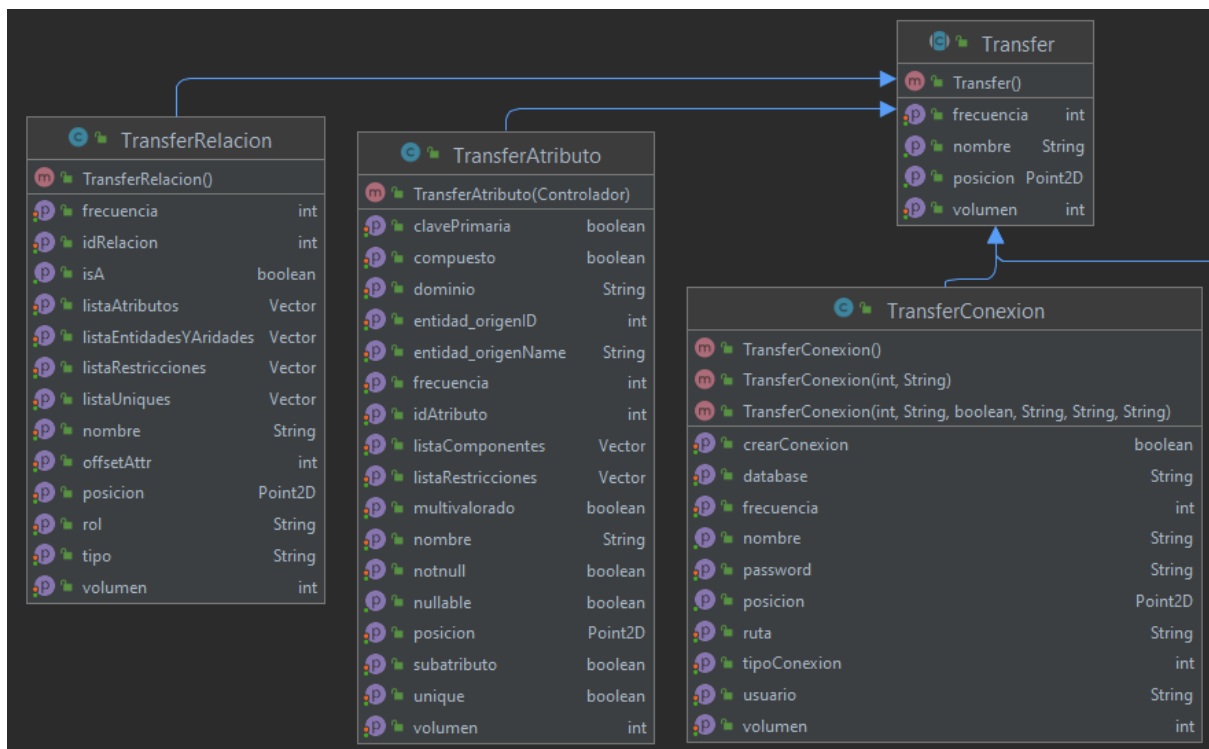
#### 3.2.1.1 Diagramas de clases paquete Transfers

Define las clases que modelan los objetos en el flujo de mensajes de la aplicación. Los objetos Transfer son aquellos que se transmiten de módulo en módulo, a partir de mensajes, y en los que se almacena cualquier cambio o acción que ha de ser tratada.

El acceso a cada uno de los objetos transfer se realiza por medio de *getters* y de *setters*.

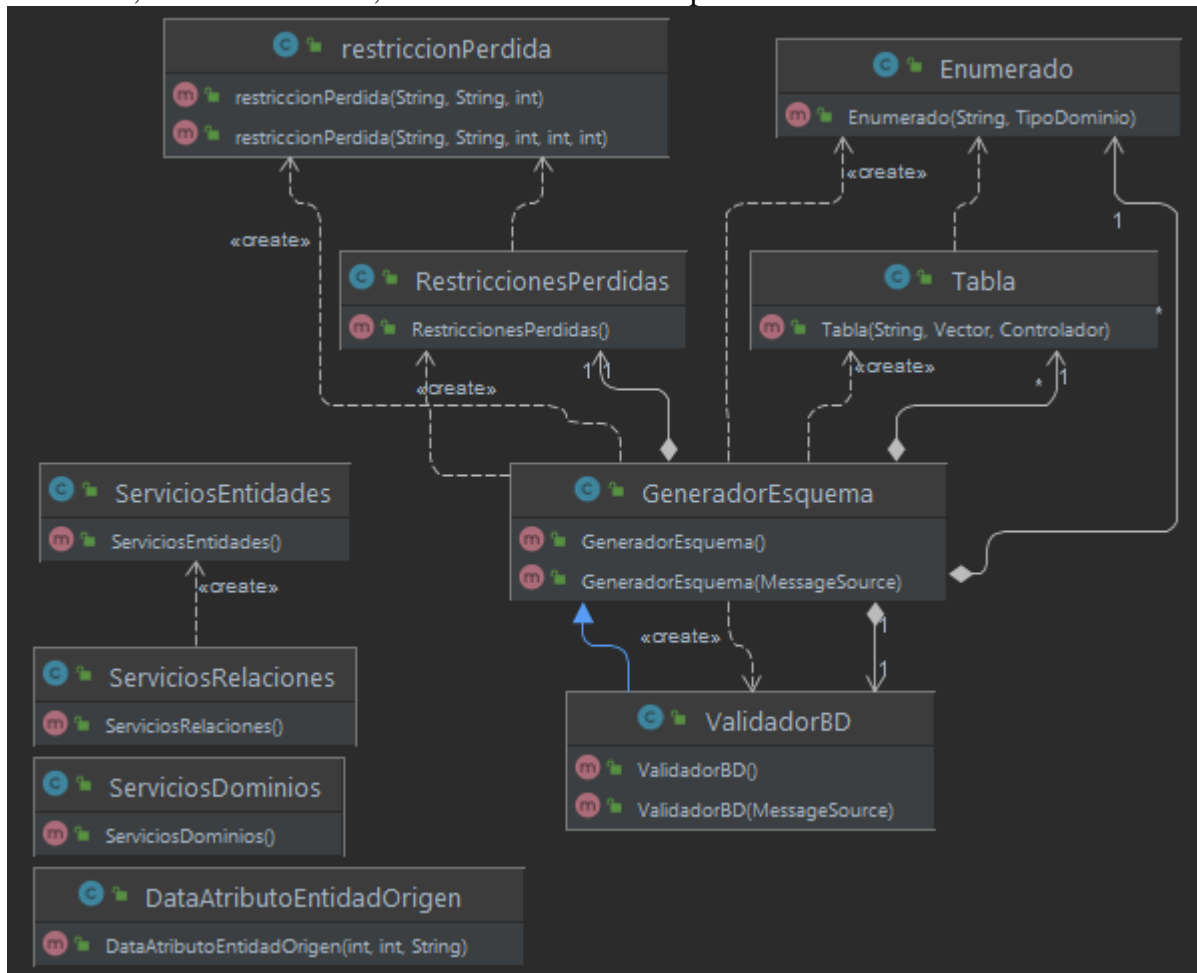
La ventaja de trabajar con objetos transfers es su propia definición: independizar el intercambio de datos entre capas. Con ellos promovemos el modularidad del sistema y la independencia entre capas del mismo.

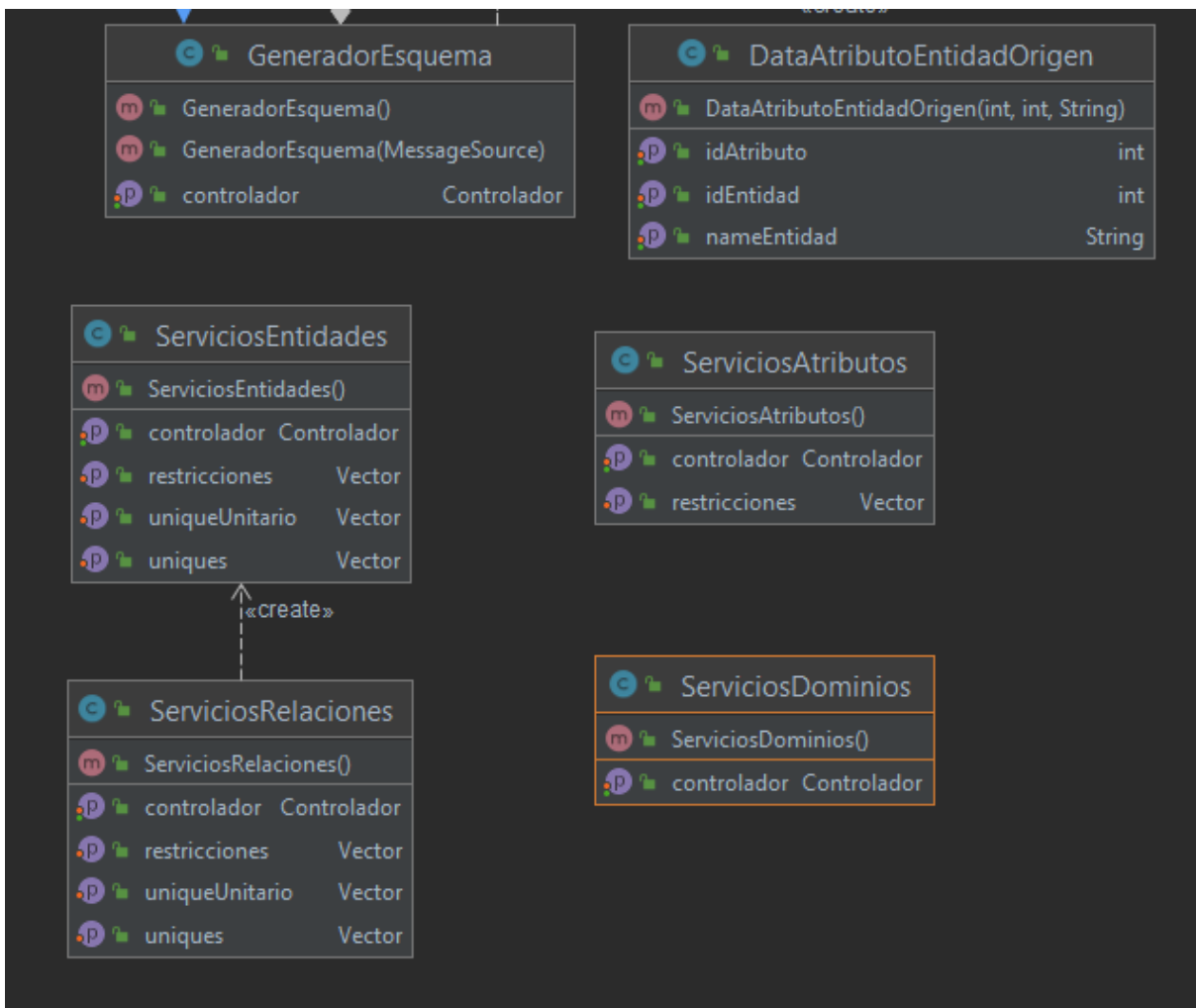
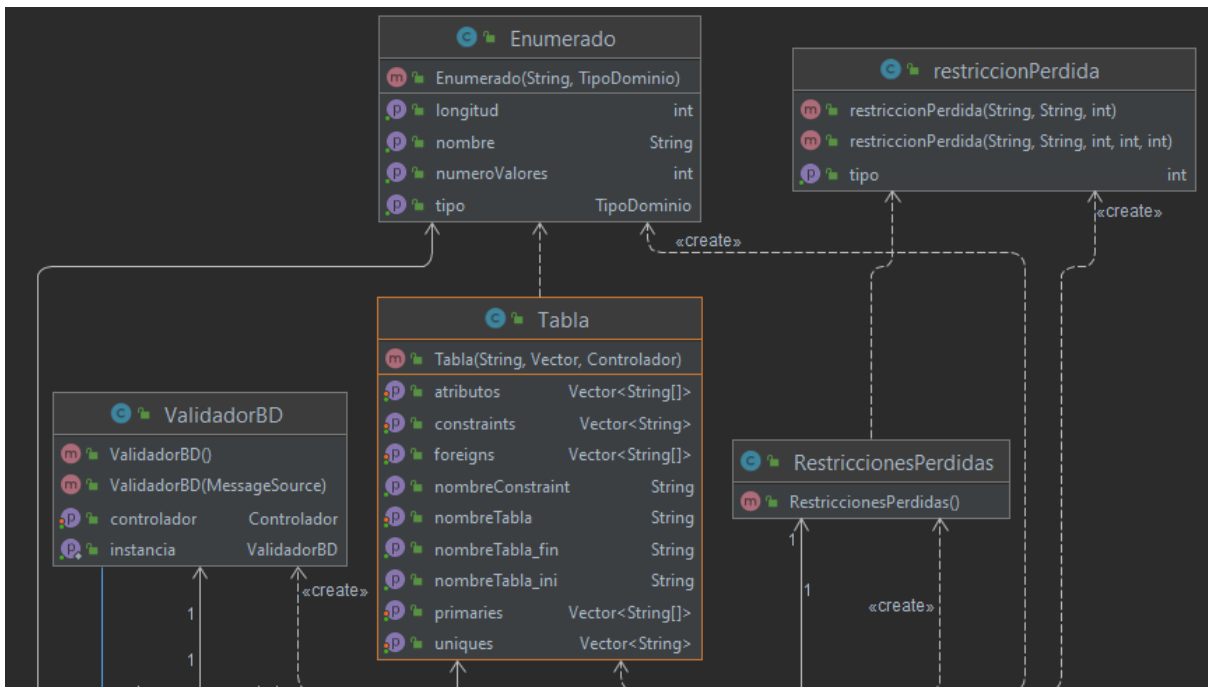




### 3.2.1.2 Diagramas de clases paquete Servicios

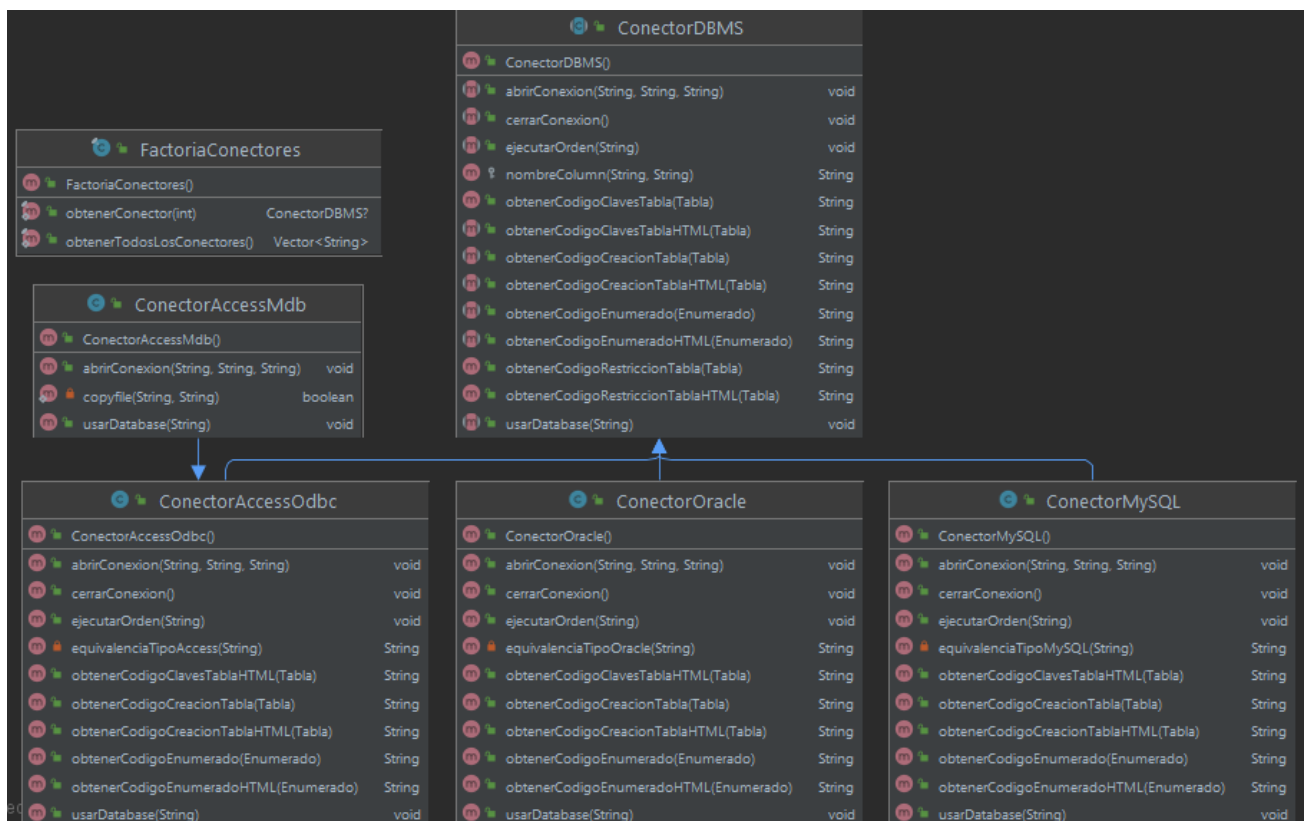
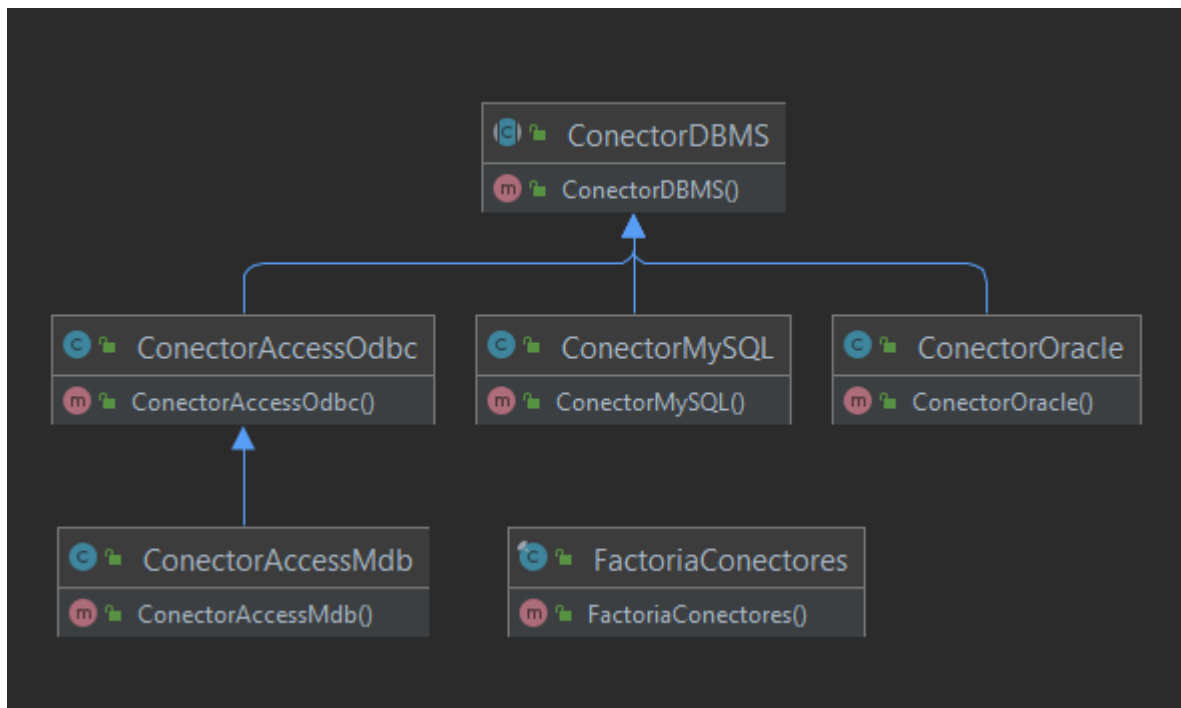
Proporciona a la aplicación el interfaz y ejecución de los servicios lógicos ofrecidos por la misma. Dichos servicios serán abstractos para el resto de módulos de la herramienta y realizarán, de forma atómica, las acciones necesarias para su correcto funcionamiento.





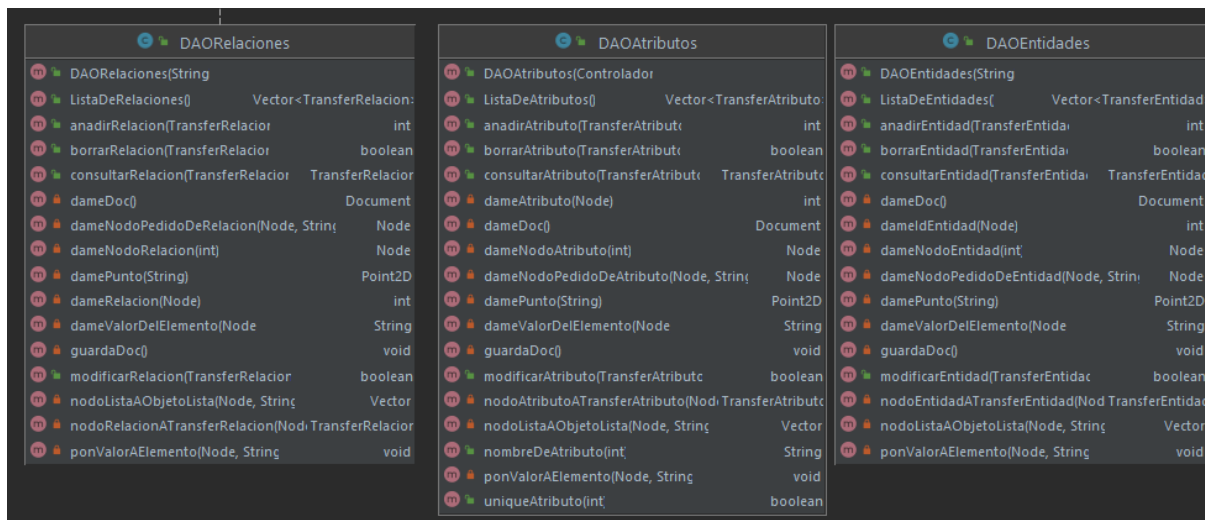
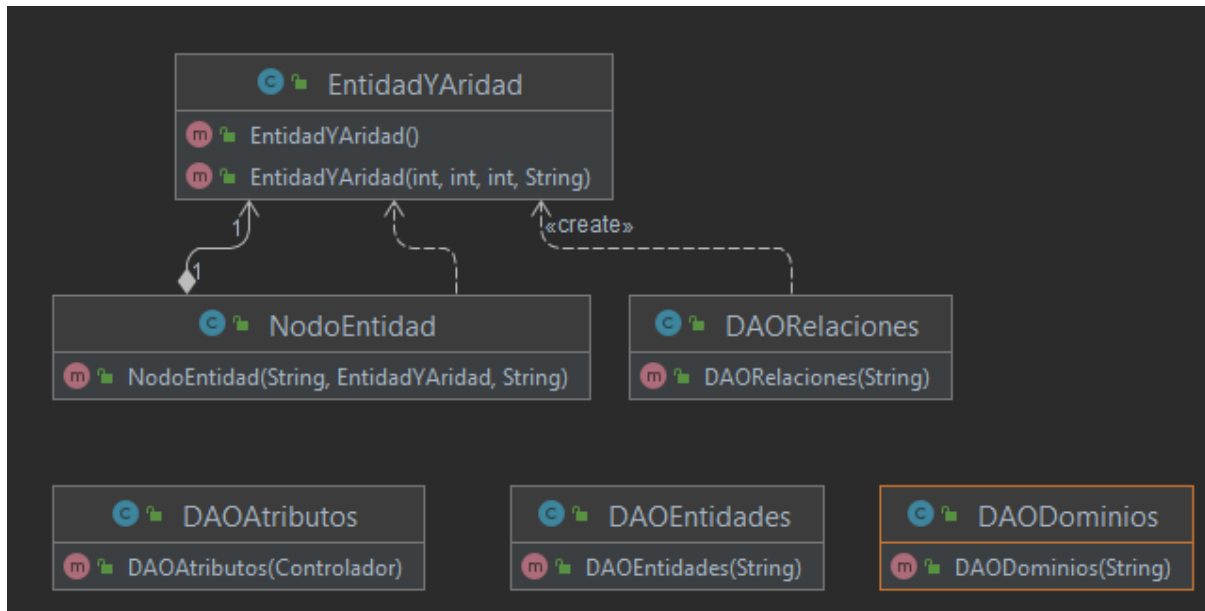
### 3.2.1.3 Diagrama de clases paquete conectorDBMS

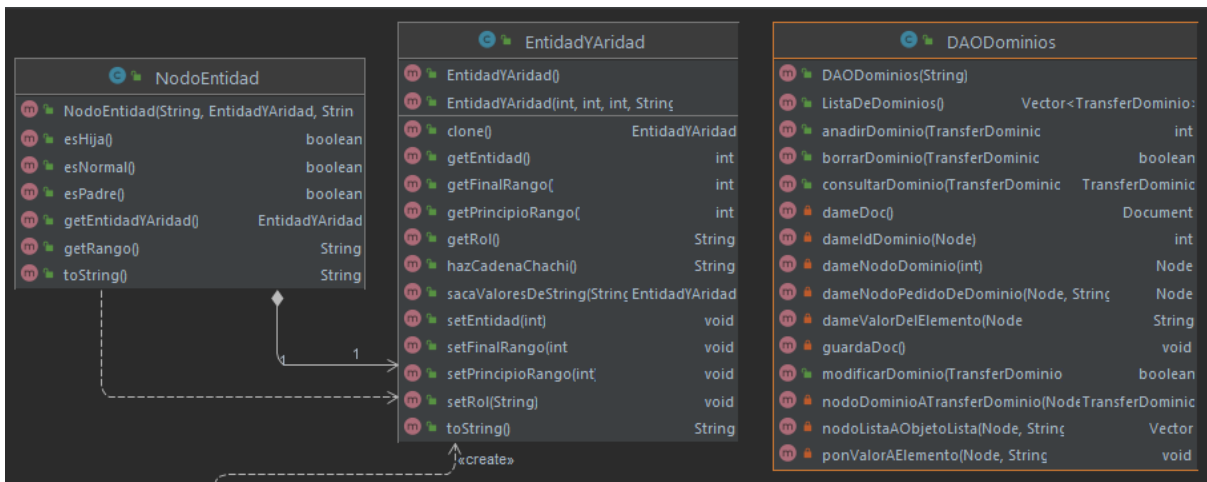
Proporciona la lógica de los diferentes conectores a las bases de datos.



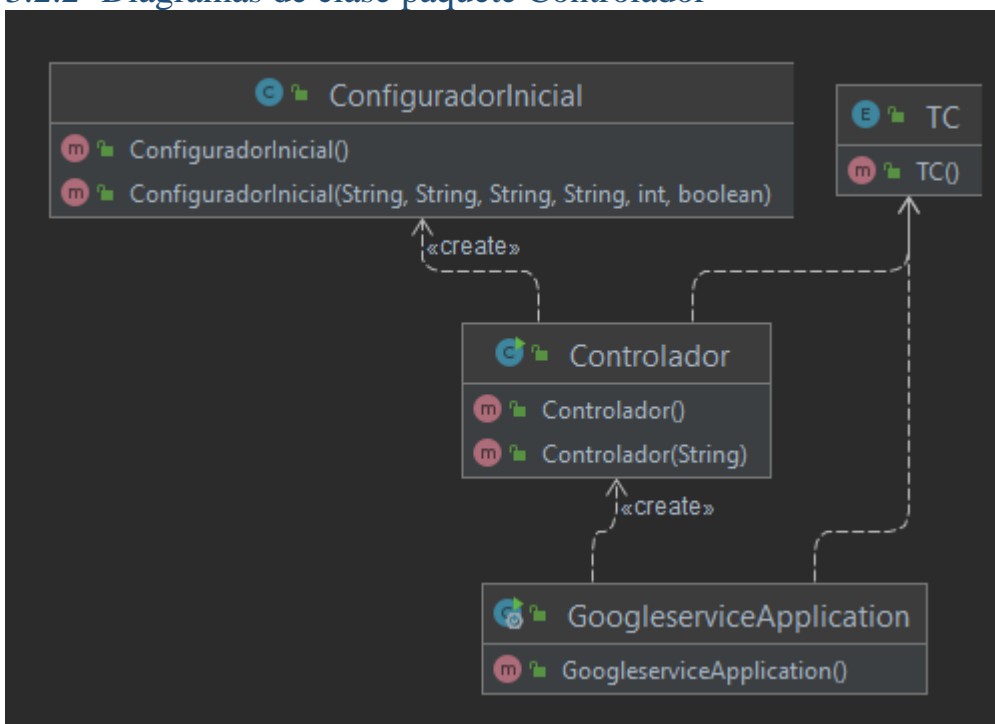
### 3.2.1.4 Diagrama de clases paquete persistencia

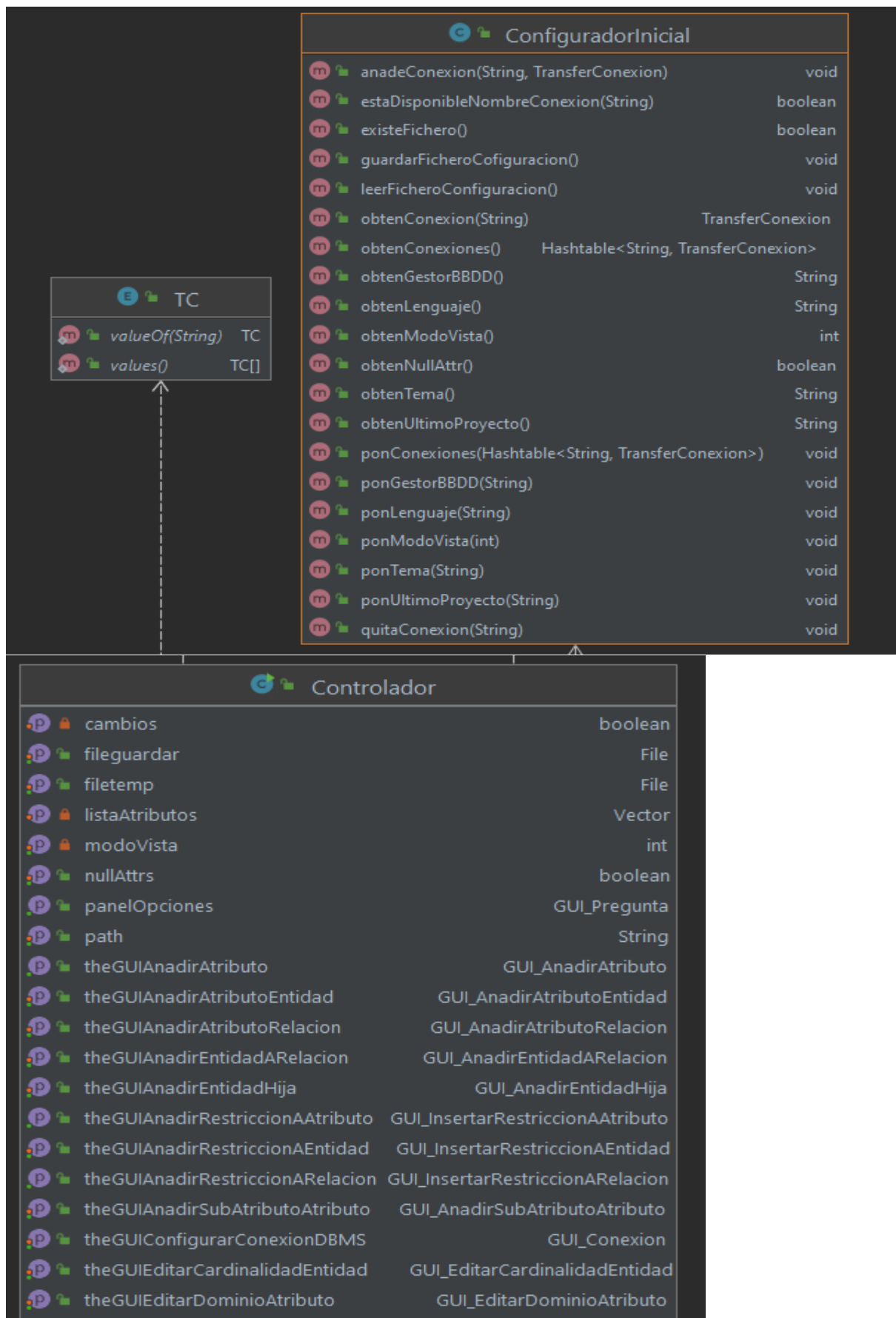
En este paquete se definen las clases que trabajan a más bajo nivel. Debido a que esta es una aplicación que trabaja con datos persistentes en todo momento estas clases son las encargadas de su correcta sincronización con el fichero temporal en disco.





### 3.2.2 Diagramas de clase paquete Controlador





theGUIInsertarDominio	GUI_InsertarDominio
theGUIInsertarEntidad	GUI_InsertarEntidad
theGUIInsertarRelacion	GUI_InsertarRelacion
theGUIModificarElementosDominio	GUI_ModificarDominio
theGUIPrincipal	GUIPrincipal
theGUIQuitarEntidadARelacion	GUI_QuitarEntidadARelacion
theGUIQuitarEntidadHija	GUI_QuitarEntidadHija
theGUIQuitarEntidadPadre	GUI_QuitarEntidadPadre
theGUIRenombrarAtributo	GUI_RenombrarAtributo
theGUIRenombrarDominio	GUI_RenombrarDominio
theGUIRenombrarEntidad	GUI_RenombrarEntidad
theGUIRenombrarRelacion	GUI_RenombrarRelacion
theGUITablaUniqueEntidad	GUI_TablaUniqueEntidad
theGUITablaUniqueRelacion	GUI_TablaUniqueRelacion
theGUIWorkSpace	GUI_SaveAs
theGuiSeleccioneionConexion	GUI_SeleccionarConexion
theGuiSeleccionarConexion	GUI_SeleccionarConexion
theServiciosAtributos	ServiciosAtributos
theServiciosDominios	ServiciosDominios
theServiciosEntidades	ServiciosEntidades
theServiciosRelaciones	ServiciosRelaciones
theServiciosSistema	GeneradorEsquema
title	String

GoogleServiceApplication	
cargarHashDeNombres(HashMap<Integer, DataAtributoEntidadOrigen>, List<Node>, List<Node>, List<Edge>, List<Edge>)	void
checkConnection(Generic, HttpServletRequest, HttpServletResponse)	String
executeQueries(Generic, HttpServletRequest, HttpServletResponse)	String
fileUpload(MultipartFile)	String
generateData(Generic, HttpServletRequest, HttpServletResponse)	String
generateDataScriptSQL(Generic, HttpServletRequest, HttpServletResponse)	String
generateEsquema(List<Node>, List<Edge>, Controlador, Boolean, HashMap<Integer, DataAtributoEntidadOrigen>)	String
generateEsquemaScriptSQL(List<Node>, List<Edge>, String, String, Controlador, Boolean, HashMap<Integer, DataAtributoEntidadOrigen>)	String
getInstrucciones(String)	String
getText(String)	String
inicio(Model, Principal, String, String)	ModelAndView
login(Model, Principal)	ModelAndView
main(String[])	void
redirectCookie(Principal, RedirectAttributes, String)	RedirectView
redirectCookieLanguage(RedirectAttributes, HttpServletResponse, String)	RedirectView
redirectCookieTheme(RedirectAttributes, HttpServletResponse, String)	RedirectView
redirectWithUsingRedirectView(RedirectAttributes)	RedirectView
updateLabelDom(String, String)	String
user(Principal)	Principal

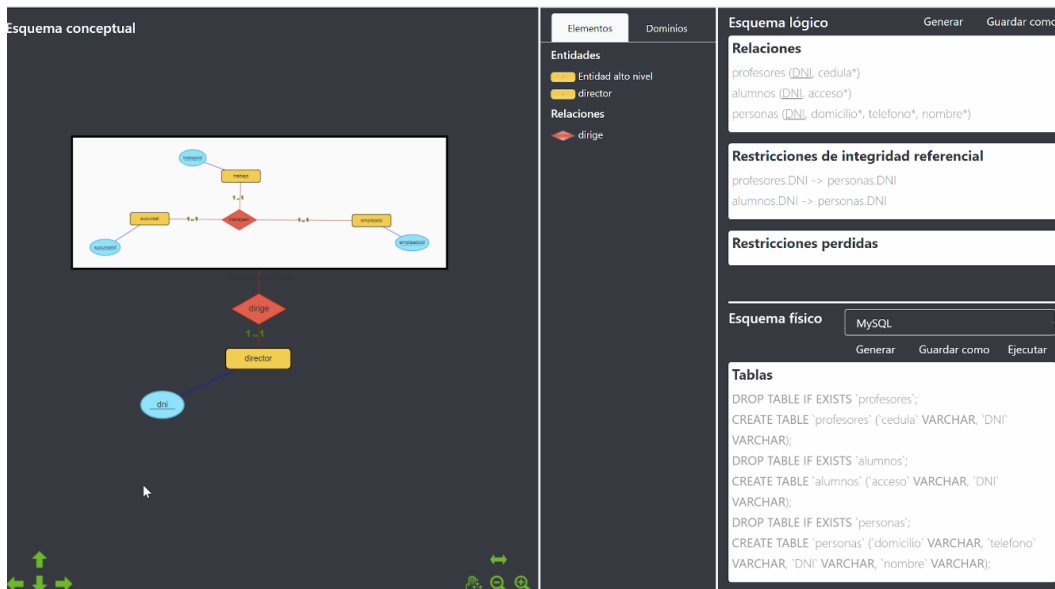
## 4 Desarrollo

La finalidad de este proyecto es ampliar y mejorar la aplicación DBCase Web.

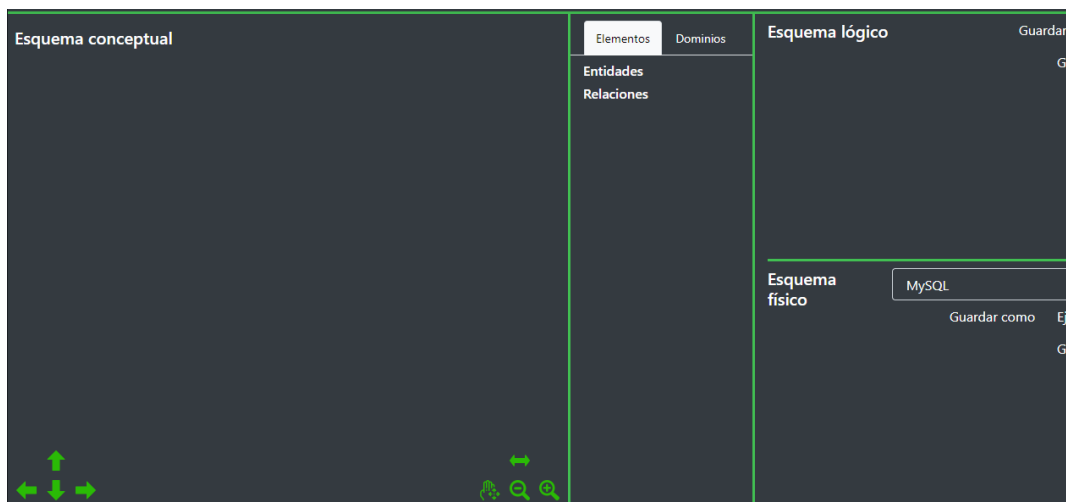
En este apartado se indicarán el desarrollo que da lugar a las aportaciones y mejoras que presenta esta versión de la aplicación.

### 4.1 Mejoras visuales.

**Contraste de colores.** Esta mejora viene de la necesidad de contrastar mejor los colores en el modo oscuro además de los colores de las cuadrículas que separan los módulos de esquema conceptual, esquema lógico, esquema físico, así como también el módulo de los elementos y dominios.

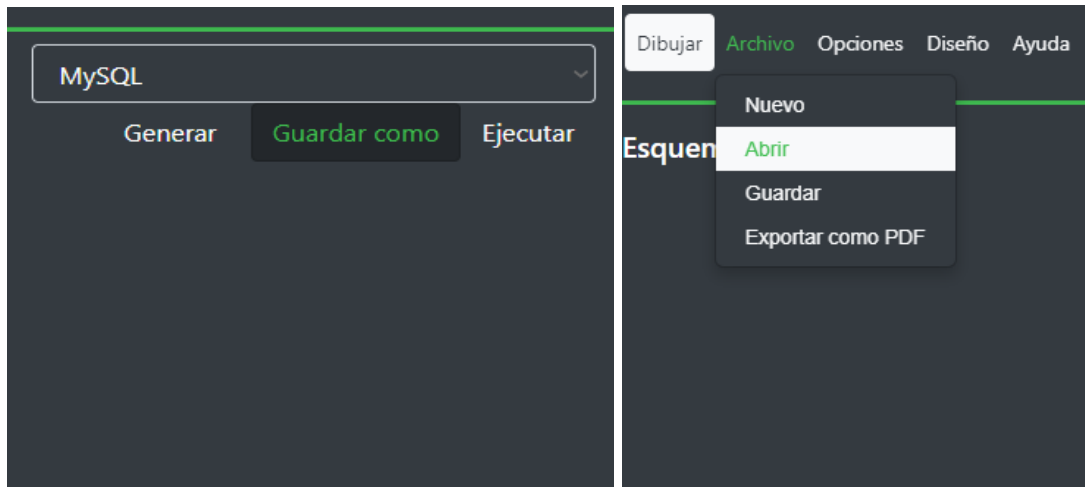


*antes*



*ahora*

·Otros elementos a los que se les aplica la mejora del contraste de color es cuando se pasa por encima a los textos del menú, o de acciones como exportar, generar. También se les aplica el mismo verde de las cuadrículas.



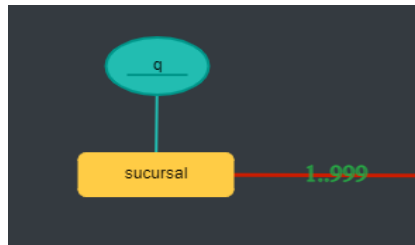
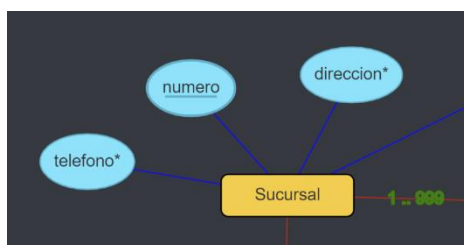
*Cambios en el fichero src/main/resources/static/css/main.css*

**Selección de nodo.** El color de los objetos dibujados (entidad, atributos, relación, IsA), varían su color respecto a la versión anterior, aunque lo que más hay que destacar es que cuando se selecciona un nodo, se sabe cuál es el nodo que se está seleccionando.



**Mejora en la visibilidad de las etiquetas en cuanto al tamaño de la letra**

*antes*



*ahora*

*Cambios en el fichero src/main/resources/static/js/vis-dates*

**Limpiar paneles del esquema lógico y físico si se quiere trabajar en un nuevo diseño.**

Cuando la aplicación tenía un esquema lógico y físico generado, y el usuario quería realizar otro diseño, los paneles del esquema lógico y físico se quedaban con los datos del diseño anterior. Se ha implementado la solución para que esto no suceda, es decir, cuando se desea generar otro diseño, ya sea nuevo o cargando un diseño ya hecho, estos paneles quedan vacíos.

The screenshot shows the DBCASE Web interface. The main window is titled "AlumnProfes .dbw". On the left, the "Esquema conceptual" panel displays a diagram with a central entity "personas" (yellow rectangle) connected to attributes "nombre", "DNI", "domicilio", and "telefono" (blue ovals). Below it, two sub-entities "alumnos" and "profesores" (yellow rectangles) are shown, both connected to "personas" via an "ISA" relationship (red triangle). The "alumnos" entity has an attribute "access". In the center, the "Elementos" panel lists "Entidades" (personas, profesores, alumnos) and "Relaciones" (ISA). On the right, the "Esquema lógico" panel shows tables: "alumnos (DNI, access)", "profesores (DNI)", and "personas (DNI, nombre, domicilio, telefono)". Below it, "Restricciones de integridad referencial" lists "alumnos.personas\_DNI -> personas.DNI" and "profesores.personas\_DNI -> personas.DNI". The "Esquema físico" panel shows the database engine set to "MySQL" and buttons for "Guardar como", "Ejecutar", and "Generar". The "Tablas" panel contains SQL code: "DROP TABLE IF EXISTS 'personas'; CREATE TABLE 'personas' ('DNI' VARCHAR, 'nombre' VARCHAR NOT NULL, 'domicilio' VARCHAR NOT NULL, 'telefono' VARCHAR NOT NULL);".

The screenshot shows the DBCASE Web interface with a different design. The main window is titled "ejemplo\_relaciones\_e.dbw". The "Esquema conceptual" panel shows entities "Telefonos" (yellow rectangle) and "Personas" (yellow rectangle) connected by a "tiene" relationship (red diamond). Below, entities "t" and "s" (yellow rectangles) are connected by an "r" relationship (red diamond). Attributes "numero" and "telefono" are connected to "Telefonos", and "id" and "apellido" are connected to "Personas". The "Elementos" panel lists "Entidades" (Telefonos, Personas, t, s) and "Relaciones" (tiene, r). The "Esquema lógico" and "Esquema físico" panels are empty, with the database engine set to "MySQL".

*src/main/resources/static/js/events.js*

## 4.2 Configuración

### Configuración para ejecutarse con cualquier cuenta ucm.es

La autenticación de la aplicación se ha hecho con la librería de Spring Boot “Spring Security OAuth 2.0 Client” y se ha configurado con el servicio “Google OAuth2 Sign-In” para poder autenticarse con la cuenta de la UCM.

The screenshot shows the Google Cloud Platform console interface. At the top, there is a navigation bar with 'Google Cloud Platform', a dropdown menu for 'DBCaseweb3', and a search bar 'Buscar productos y recursos'. Below this, the breadcrumb trail reads 'API > API y servicios > ID de cliente para Aplicación web'. On the right side of the breadcrumb, there are links for 'DESCARGAR JSON' and 'RESTABLECE'. The left sidebar contains a menu with items: 'Panel', 'Biblioteca', 'Credenciales' (highlighted), 'Pantalla de consentimiento ...', 'Verificación del dominio', and 'Acuerdos de uso de páginas'. The main content area is titled 'ID de cliente para Aplicación web' and contains a form for configuring an OAuth 2.0 client. The 'Nombre \*' field is filled with 'DBCCase2.0'. Below the field, there is a note: 'El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.' An information box states: 'Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu pantalla de consentimiento de OAuth como dominios autorizados.' Below this, the section 'Orígenes autorizados de JavaScript' is shown with the subtext 'Para usar con solicitudes de un navegador'. There are two 'URI \*' input fields, the first containing 'https://localhost:8080' and the second containing 'http://localhost:8080'.

## URI de redireccionamiento autorizados

Para usar con solicitudes de un servidor web

URI \*



+ AGREGAR URI

### 4.3 Solución de problemas con la traducción de esquemas lógico y físico

Cuando se generaba el modelo físico para MySQL, se imponían primero restricciones de integridad referencial sobre una clave primaria que aún no está definida y por tanto daba errores al ejecutar el script con la BD.

```
ALTER TABLE `alumnos` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `alumnos` ADD FOREIGN  
KEY (`DNI`) REFERENCES personas(`DNI`);  
ALTER TABLE `profesores` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `profesores` ADD FOREIGN  
KEY (`DNI`) REFERENCES personas(`DNI`);  
ALTER TABLE `personas` ADD PRIMARY KEY (`DNI`);
```

Pasa a ser

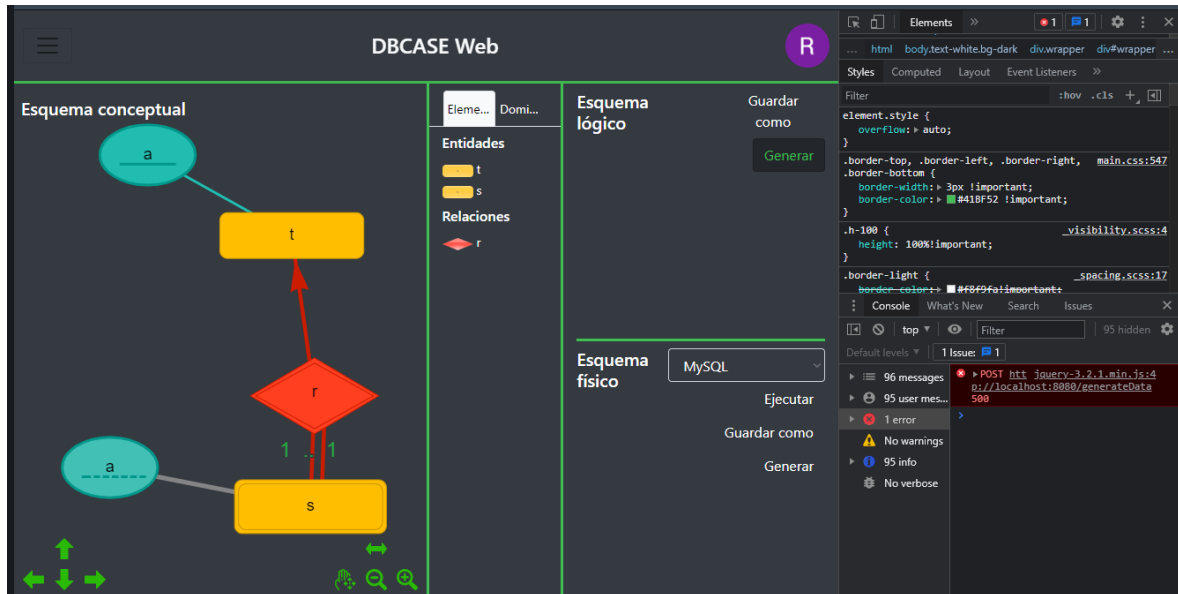
---

#### Claves

```
ALTER TABLE `personas` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `alumnos` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `alumnos` ADD FOREIGN KEY (`DNI`) REFERENCES personas(`DNI`);  
ALTER TABLE `profesores` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `profesores` ADD FOREIGN KEY (`DNI`) REFERENCES personas(`DNI`);
```

---

-Al intentar generar el esquema físico y lógico de entidades débiles fallaba



Cambios en los fichero:

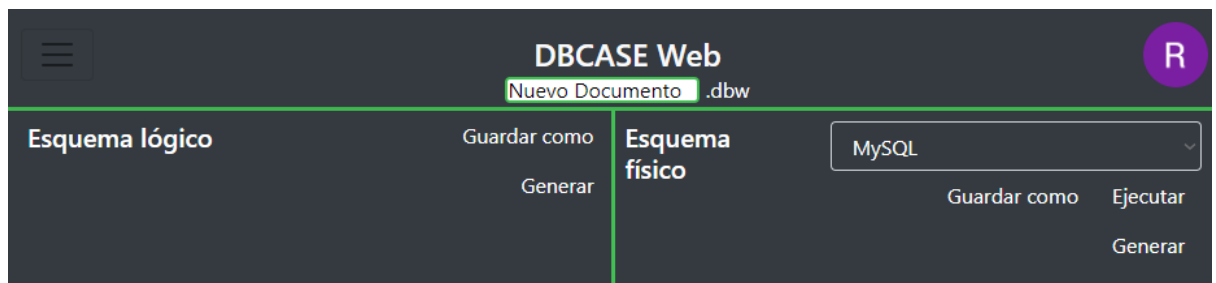
`src/main/java/modelo/servicios/GeneradorEsquema.java`

`src/main/java/modelo/servicios/Tabla.java`

## 4.4 Nuevas funcionalidades

**Permitir al usuario ver el nombre del archivo con el que se está trabajando o “nuevo documento” en el caso de ser un nuevo documento.**

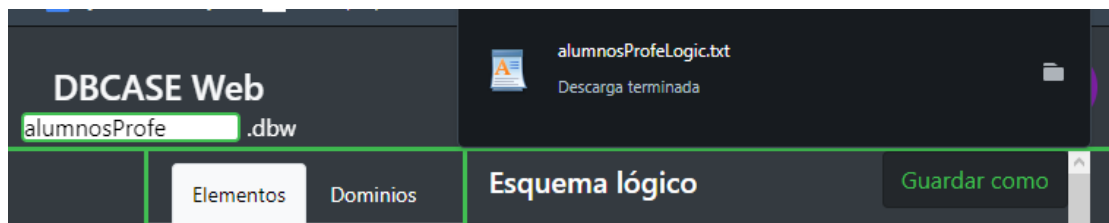
Se ha agregado una casilla de texto donde o bien indica que es un documento nuevo, o bien indica el nombre del documento con el que se está trabajando actualmente. Al descargar el esquema lógico, se agrega al nombre del documento la palabra clave “Logic”, al descargar el esquema físico se agrega la palabra clave “Physic” y para descargar en pdf se agrega la palabra clave “General”.



*por defecto*



*al abrir un documento*



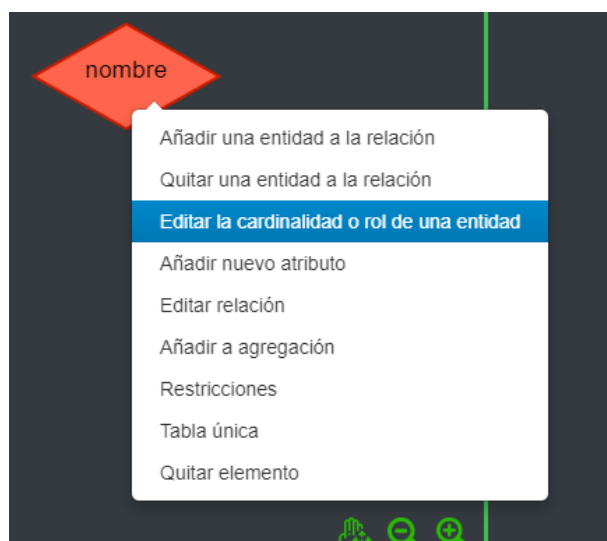
*al descargar un esquema lógico*

*Cambios en los ficheros:*

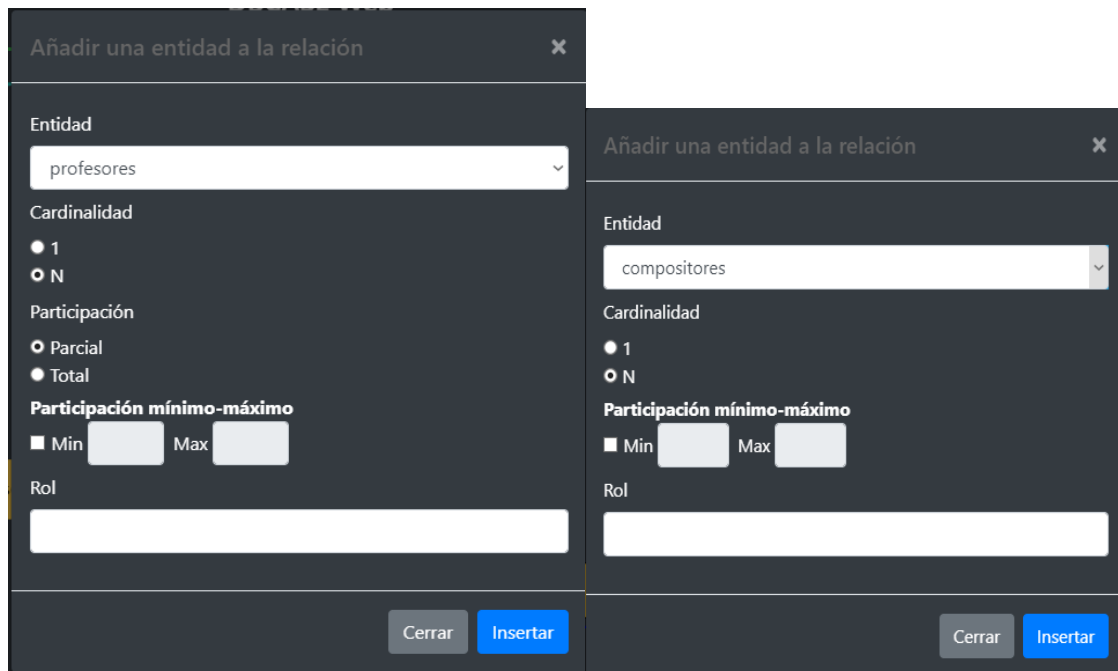
- src/main/resources/static/js/saveAndLoads.js*
- src/main/resources/templates/inicio.html*
- src/main/resources/language/messages.properties*
- src/main/resources/language/messages\_en.properties*

### **Implementación de participación total o parcial en la pantalla de cardinalidad y participación y corrección de errores.**

Se ha modificado y corregido errores de funcionamiento en el modal “Añadir entidad a la relación” y del modal “Editar cardinalidad o rol de una entidad”.



En esta versión al añadir o editar una entidad a la relación, se puede elegir también la participación.

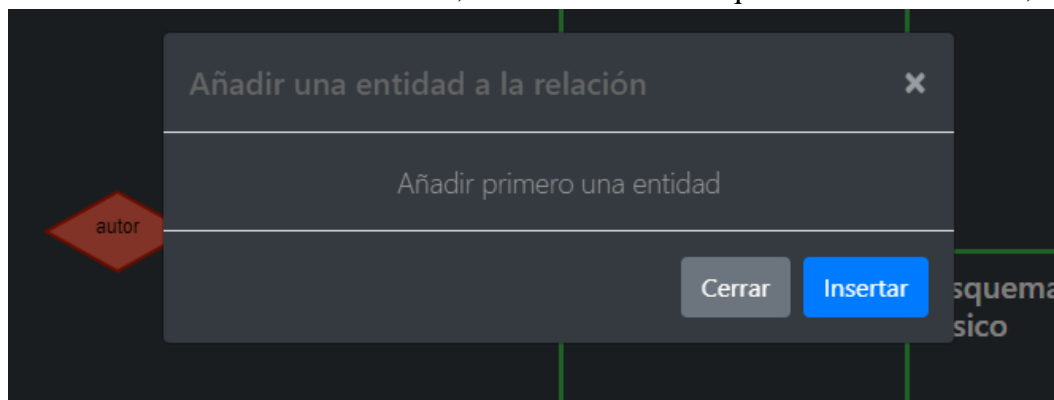


*Ahora*

*Antes*

Este modal esta hecho rigurosamente para que no exista una traducción equivocada de la participación mínimo máximo independientemente de la cardinalidad o participación seleccionada. Se especifican los siguientes casos.

- Al intentar abrir este modal, si no existe entidad que añadir a la relación, se indica



Algo parecido sucede cuando se intenta editar una cardinalidad o rol de alguna entidad y que dicha relación no tenga ninguna entidad agregada a la cual editar.

- Al abrir este modal, por defecto las opciones son Cardinalidad N, con participación parcial, sin participación mínimo-máximo y con un rol en blanco

Entidad: entidad1

Cardinalidad:  
 1  
 N

Participación:  
 Parcial  
 Total

Participación mínimo-máximo:  
 Min  Max

Rol:

Cerrar Insertar

- Si se selecciona participación Total, se activa la participación mínimo-máximo, colocando, el mínimo como 1 y el máximo como N, valores por defecto. Si el usuario deja en blanco alguno de los valores, min/max, se rellenan automáticamente con sus valores por defecto siempre y cuando hay una participación total marcada y el checkbox de participación mínimo-máximo también está marcado. Si dicho checkbox se desmarca, los valores de los campos min/max quedan en blanco y dichos campos no son editables.

Entidad: entidad1

Cardinalidad:  
 1  
 N

Participación:  
 Parcial  
 Total

Participación mínimo-máximo:  
 Min 1 Max N

Rol:

Cerrar Insertar

El mismo comportamiento ocurre cuando se marca una participación parcial, con la única diferencia que su valor por defecto en el min es 0 y que este campo no se puede editar para seguir con la coherencia de la teoría de las bases de datos relacionales.

DBCASE Web

Añadir una entidad a la relación

Entidad

entidad1

Cardinalidad

1

N

Participación

Parcial

Total

Participación mínimo-máximo

Min 0 Max N

Rol

- Cuando se desea añadir una relación recursivamente, el programa detecta si esa entidad ya está asociada a la relación y rellena automáticamente el campo de la etiqueta del rol con “ROL”.

Ayuda

DBCASE Web

Añadir una entidad a la relación

Entidad

entidad1

Cardinalidad

1

N

Participación

Parcial

Total

Participación mínimo-máximo

Min Max

Rol

ROL

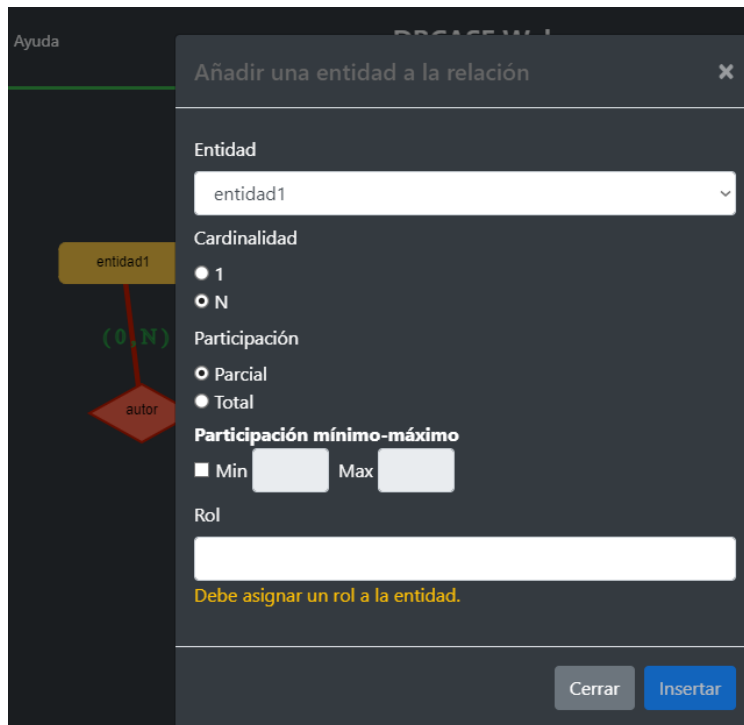
Cerrar Insertar

entidad1

(0,N)

autor

- Si dicha etiqueta es eliminada, se exige una etiqueta para el rol o de otra manera no se procederá a guardar esta recursión.



- Se ha implementado también deshabilitar los atajos de teclado, como Supr. Ya que, si se pulsaba esta tecla al escribir en un textbox como Rol, se borraba el elemento gráfico que estuviese seleccionado en el lienzo.

*Cambios en los ficheros:*

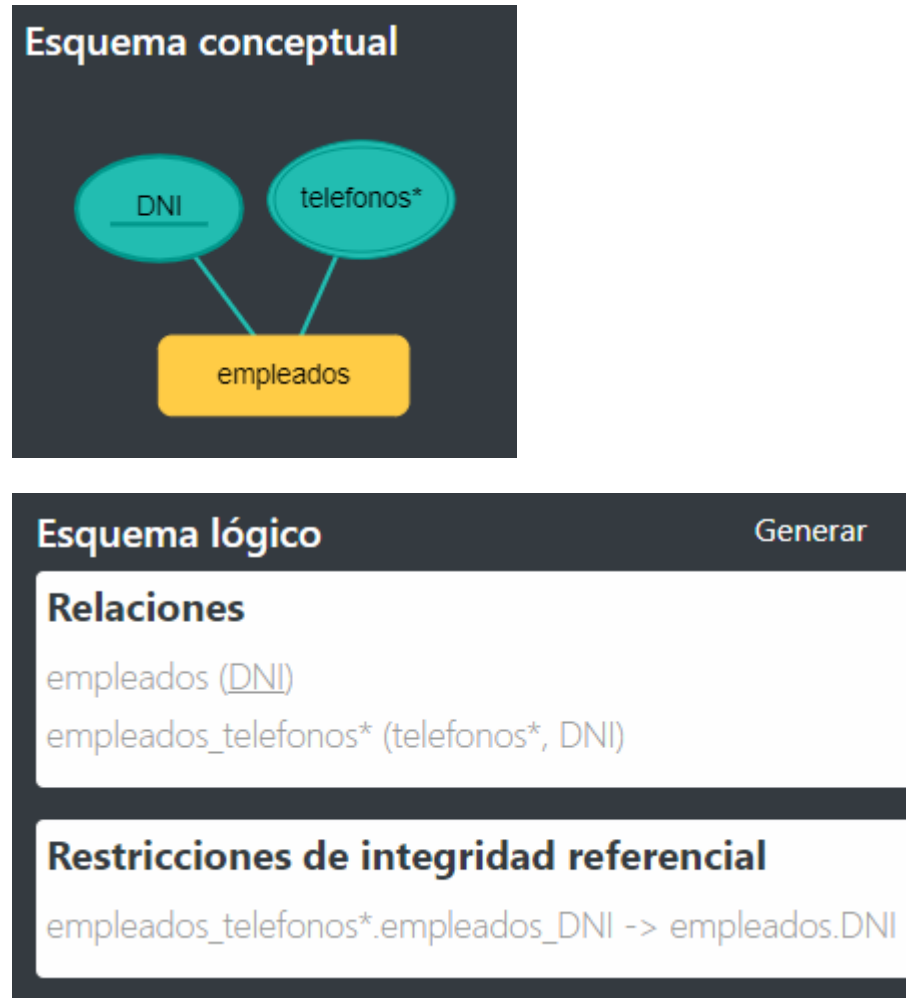
*src/main/resources/static/js/events.js*

*src/main/resources/templates/inicio.html*

## 4.5 Ejemplos de traducción

A continuación, se muestran algunos ejemplos de traducción lógica y física de varios diseños.

### 4.5.1 Atributos multivalorados



## Esquema físico

MySQL

Generar Guardar como Ejecutar

### Tablas

```
DROP TABLE IF EXISTS `empleados`;  
CREATE TABLE `empleados` (`DNI` VARCHAR);  
DROP TABLE IF EXISTS `empleados_telefonos`;  
CREATE TABLE `empleados_telefonos` (`telefonos` VARCHAR, `DNI`  
VARCHAR);
```

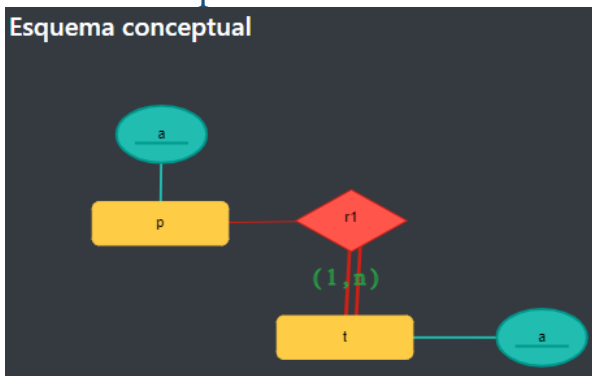
### Tipos enumerados

### Claves

```
ALTER TABLE `empleados` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `empleados_telefonos` ADD FOREIGN KEY  
(`empleados_DNI`) REFERENCES empleados(`DNI`);
```

## 4.5.2 Participación total

### Esquema conceptual



## Esquema lógico

### Relaciones

p (a)

t (a)

r1 (p\_a, t\_a)

### Restricciones de integridad referencial

r1.p\_a -> p.a

r1.t\_a -> t.a

## Esquema físico

MySQL

Generar

Guardar como

### Tablas

```
DROP TABLE IF EXISTS `p`;
```

```
CREATE TABLE `p` (`a` VARCHAR);
```

```
DROP TABLE IF EXISTS `t`;
```

```
CREATE TABLE `t` (`a` VARCHAR);
```

```
DROP TABLE IF EXISTS `r1`;
```

```
CREATE TABLE `r1` (`p_a` VARCHAR, `t_a` VARCHAR NOT NULL);
```

### Tipos enumerados

### Claves

```
ALTER TABLE `t` ADD PRIMARY KEY (`a`);
```

```
ALTER TABLE `p` ADD PRIMARY KEY (`a`);
```

```
ALTER TABLE `r1` ADD PRIMARY KEY (`p_a`, `t_a`);
```

```
ALTER TABLE `r1` ADD FOREIGN KEY (`p_a`) REFERENCES p(`a`);
```

```
ALTER TABLE `r1` ADD FOREIGN KEY (`t_a`) REFERENCES t(`a`);
```

### 4.5.3 Entidad débil

**Esquema conceptual**



**Esquema lógico** Ge

**Relaciones**

`s (s_a, t_a)`  
`t (a)`

**Restricciones de integridad referencial**

`s.t_a -> t.a`

**Esquema físico** MySQL Generar Gua

**Tablas**

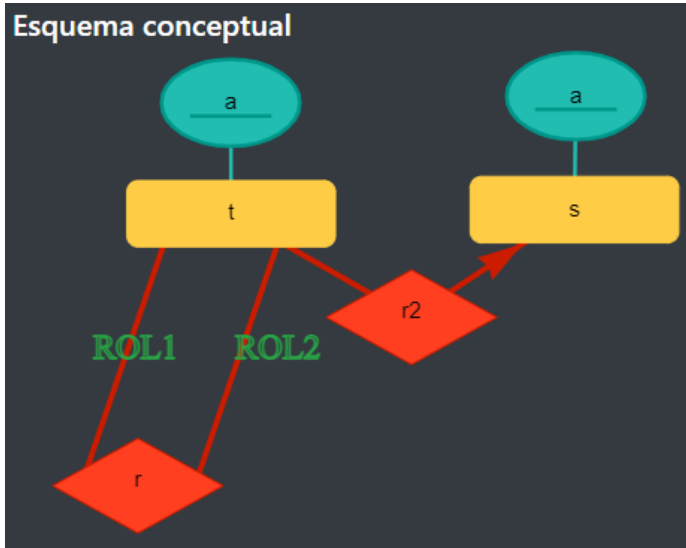
```
DROP TABLE IF EXISTS `s`;  
CREATE TABLE `s` (`s_a` VARCHAR, `t_a` VARCHAR);  
DROP TABLE IF EXISTS `t`;  
CREATE TABLE `t` (`a` VARCHAR);
```

**Tipos enumerados**

**Claves**

```
ALTER TABLE `t` ADD PRIMARY KEY (`a`);  
ALTER TABLE `s` ADD PRIMARY KEY (`s_a`, `t_a`);  
ALTER TABLE `s` ADD FOREIGN KEY (`t_a`) REFERENCES t(`a`);
```

#### 4.5.4 Relación recursiva



#### Esquema lógico

##### Relaciones

$s(a)$

$t(a)$

$r2(t_a, s_a)$

$r(ROL2_a, ROL1_a)$

##### Restricciones de integridad referencial

$r2.t_a \rightarrow t.a$

$r2.s_a \rightarrow s.a$

$r.ROL2_a \rightarrow t.a$

$r.ROL1_a \rightarrow t.a$

### Tablas

```
DROP TABLE IF EXISTS `s`;  
CREATE TABLE `s` (`a` VARCHAR);  
DROP TABLE IF EXISTS `t`;  
CREATE TABLE `t` (`a` VARCHAR);  
DROP TABLE IF EXISTS `r2`;  
CREATE TABLE `r2` (`t_a` VARCHAR, `s_a` VARCHAR);  
DROP TABLE IF EXISTS `r`;  
CREATE TABLE `r` (`ROL2_a` VARCHAR, `ROL1_a` VARCHAR);
```

### Claves

```
ALTER TABLE `t` ADD PRIMARY KEY (`a`);  
ALTER TABLE `s` ADD PRIMARY KEY (`a`);  
ALTER TABLE `r2` ADD PRIMARY KEY (`t_a`);  
ALTER TABLE `r2` ADD FOREIGN KEY (`t_a`) REFERENCES t(`a`);  
ALTER TABLE `r2` ADD FOREIGN KEY (`s_a`) REFERENCES s(`a`);  
ALTER TABLE `r` ADD PRIMARY KEY (`ROL2_a`, `ROL1_a`);  
ALTER TABLE `r` ADD FOREIGN KEY (`ROL2_a`) REFERENCES t(`a`);  
ALTER TABLE `r` ADD FOREIGN KEY (`ROL1_a`) REFERENCES t(`a`);
```

## 4.5.5 Relación isA

### Esquema conceptual



### Esquema lógico

#### Relaciones

profesor (DNI)

alumnos (DNI, acceso\*)

personas (DNI, nombre\*)

#### Restricciones de integridad referencial

profesor.personas\_DNI -> personas.DNI

alumnos.personas\_DNI -> personas.DNI

**Tablas**

```
DROP TABLE IF EXISTS `personas`;  
CREATE TABLE `personas` (`DNI` VARCHAR, `nombre` VARCHAR);  
DROP TABLE IF EXISTS `profesor`;  
CREATE TABLE `profesor` (`DNI` VARCHAR);  
DROP TABLE IF EXISTS `alumnos`;  
CREATE TABLE `alumnos` (`acceso` VARCHAR, `DNI` VARCHAR);
```

**Claves**

```
ALTER TABLE `personas` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `profesor` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `profesor` ADD FOREIGN KEY (`personas_DNI`) REFERENCES  
personas(`DNI`);  
ALTER TABLE `alumnos` ADD PRIMARY KEY (`DNI`);  
ALTER TABLE `alumnos` ADD FOREIGN KEY (`personas_DNI`) REFERENCES  
personas(`DNI`);
```

## 5 Conclusiones y trabajo futuro

El objetivo del proyecto fue desde un principio mejorar la versión que había y ampliar la funcionalidad. En general me siento satisfecha porque el objetivo base ha sido alcanzado: dejar una mejor aplicación de la que heredé.

La primera fase del proyecto, familiarizarme con la aplicación y testarla al mismo tiempo, fue sencillo, pues lo que más esfuerzo me llevó fue repasar las bases teóricas del modelo relacional, luego al testar la aplicación comencé a darme cuenta de lo útil que es. Añoré haber tenido esta herramienta cuando comencé a adquirir conocimientos sobre el tema de las bases de datos y su diseño.

Al comenzar la fase del desarrollo se me complicó mucho compilar el proyecto, de hecho, invertí mucho tiempo y no conseguí resultados hasta que el compañero, Luis Eduardo Paucar Cerrón, participante de la versión previa a este proyecto, colaboró conmigo para configurar en el IDE Eclipse, además me dio un pequeño “*tour*” por el interior de la aplicación y de cómo está estructurada. Un par de meses más tarde probé a compilar el proyecto en el IDE IntelliJ IDEA y fue más sencillo a diferencia de hacerlo con Eclipse.

Las primeras modificaciones fueron en la parte visual, además de repasar canvas para los gráficos, elegí colores y probé las combinaciones más adecuadas para ponerlo más cómodo a las personas que tienen dificultades visuales.

La funcionalidad nueva y reestructuración del comportamiento del módulo de cardinalidad y participación en una relación, me hizo aprender más a fondo JavaScript y jQuery. En general estoy muy contenta, pues mi conocimiento de páginas web era muy básico al comenzar este proyecto. Sin duda este proyecto ha sido un camino de aprendizaje continuo.

## 5.1 Trabajo futuro

### 5.1.1 Funcionalidades futuras

- **Funcionalidad selección múltiple o individual de elemento.** al dejar de arrastrar un grupo o elemento se debería mantener la selección, por si después de soltar el botón se vuelve a pulsar para arrastrar de nuevo.
- **Constraint en MySQL.** Este lenguaje requiere añadir como CONSTRAINT una segunda restricción de integridad referencial. Hay que añadirla como CONSTRAINT y con nombre.
- **Deshacer-Rehacer (Ctrl-Z, Ctrl-Y).** Permitir al usuario poder moverse entre las diferentes acciones realizadas en la creación del diagrama
- **Corrección en la traducción.** Algunas traducciones en la tabla no corresponde a las restricciones de integridad lo mismo pasa al traducir al esquema físico

## 6 Repositorio

El repositorio donde se encuentra el proyecto es

<https://github.com/8Roxyrox8/DBCcase3.0>

## 7 Bibliografía

- Apuntes de Bases de Datos por Javier Arroyo Yolanda García Virginia Francisco Gilmartín Iván Martínez Fernando Sáenz
- Memorias antiguas versiones DBCASE Alberto Milán Gutierrez, Miguel Martinez Segura, Francisco Javier Cáceres González, Rodrigo Denis Cepeda Mateos, Cristina Marco de Francisco, Tello Serrano Gordillo, Miguel Arriba García, Luis Eduardo Paucar Cerrón, Yrving David Conde Cubas.
- Documentación y ejemplos de HTML, JQuery, CSS, Javascript, JSON, Vis js  
<https://www.w3schools.com/>  
<https://www.geeksforgeeks.org>  
<https://developer.mozilla.org/en-US/docs/Learn/Accessibility>  
<https://www.freecodecamp.org>  
<https://visjs.github.io/vis-network/docs/network/>
- Documentación de JQuery  
<https://api.jquery.com/>
- Documentación de JSON  
<https://jsonapi.org/>
- Documentación Gson  
<https://www.javadoc.io/doc/com.google.code.gson/gson/2.8.0/com/google/gson/Gson.html>  
<https://sites.google.com/site/gson/gson-user-guide>
- Documentación de Bootstrap 4  
<https://getbootstrap.com/docs/4.1/>
- Buscador universal de problemas inusuales  
<https://stackoverflow.com>

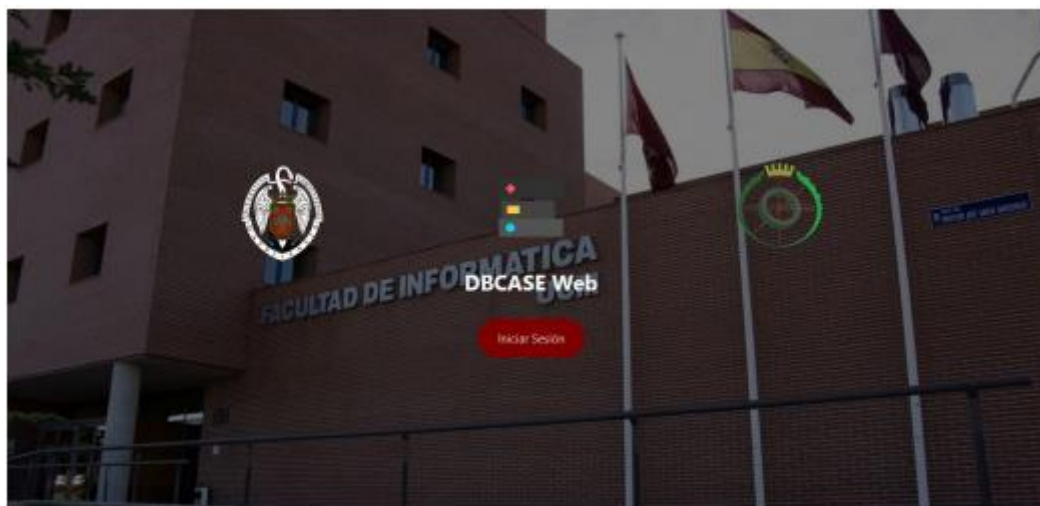
## 8 Apéndice - Manual de usuario

Para poder usar DBCase web, sólo hace falta un navegador y tener una cuenta UCM.

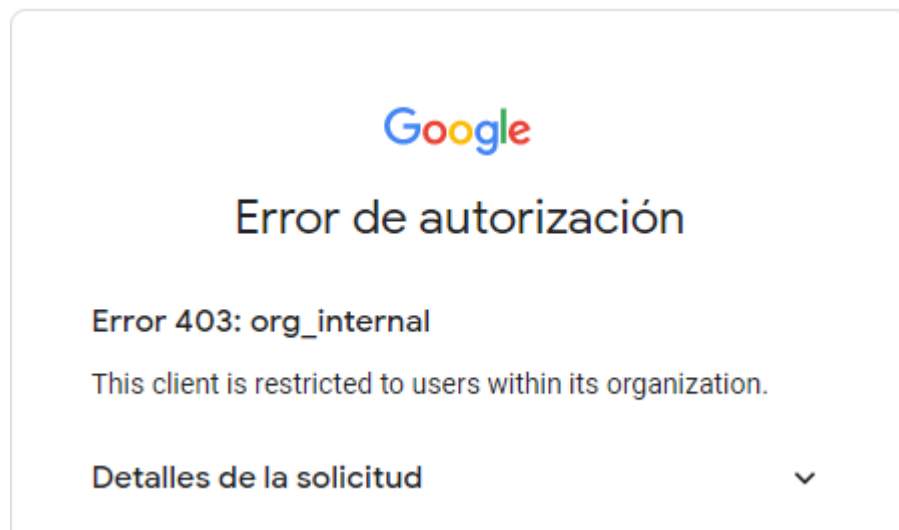
A continuación, se explica cómo trabajar con esta herramienta, comenzando por el inicio de sesión y terminando por generar esquemas después del diseño Entidad-Relación.

### 8.1 Inicio de sesión

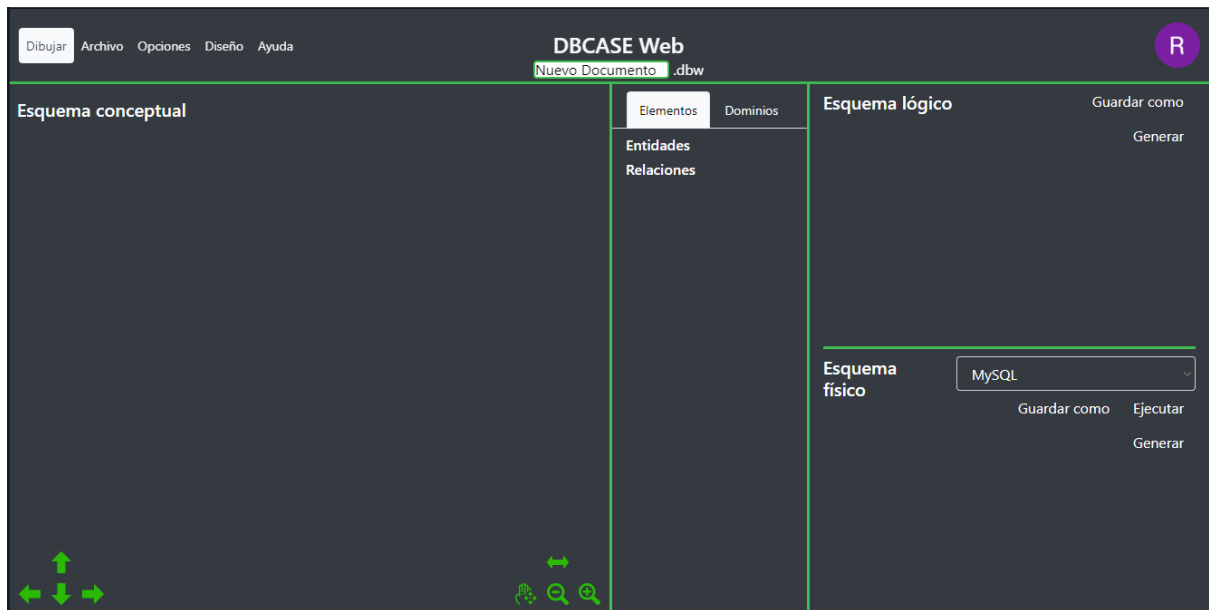
Para poder utilizar la herramienta, el programa cuenta con un inicio de sesión, hay que tener cuenta UCM. Se mostrará una página de inicio como la siguiente



Al hacer click en “*Iniciar Sesión*”, se mostrará un gestor de cuentas Google, que nos permite autenticarnos al seleccionar nuestro usuario. Si se selecciona un usuario que no pertenece a la UCM, se muestra lo siguiente:

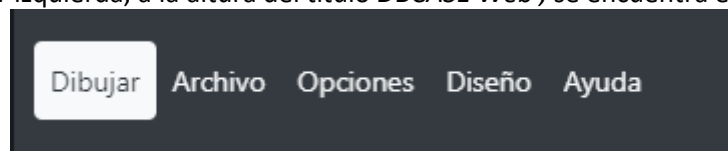


De lo contrario, si nos autentificamos con una cuenta UCM, se accede a DBCase web: Por defecto el idioma es español y el tema oscuro, con el *diseño Esquema conceptual*, *Panel de Información*, y en una misma cuadrícula dividida en dos, *Esquema lógico* y *Esquema físico*.

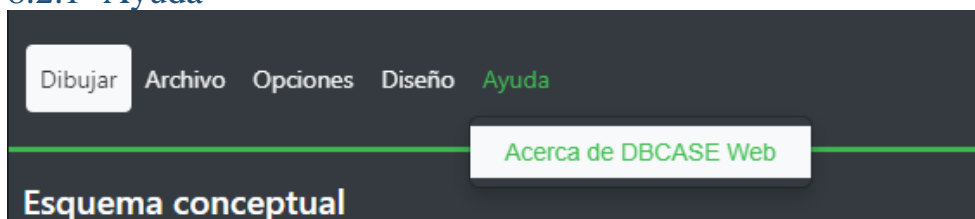


## 8.2 Menú

En la parte superior izquierda, a la altura del título *DBCASE Web*, se encuentra el menú

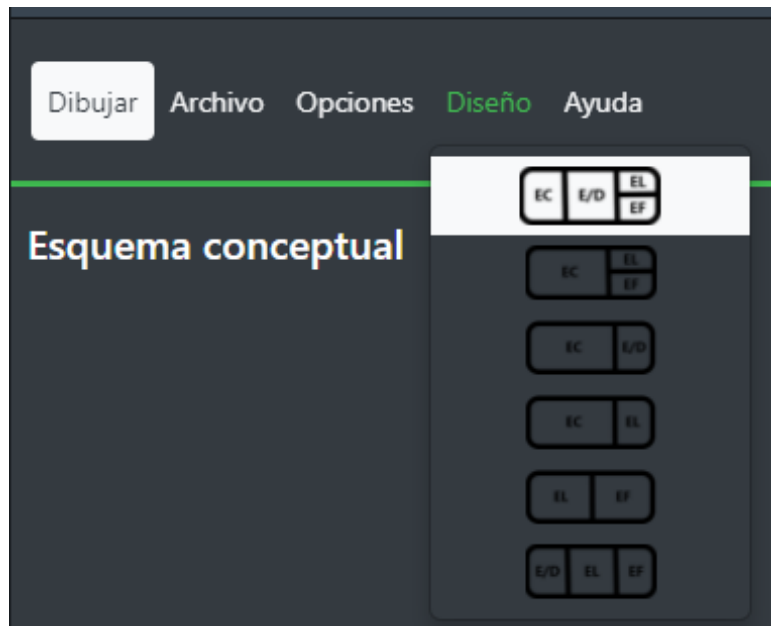


### 8.2.1 Ayuda



Ofrece información del sistema, como autores y versión.

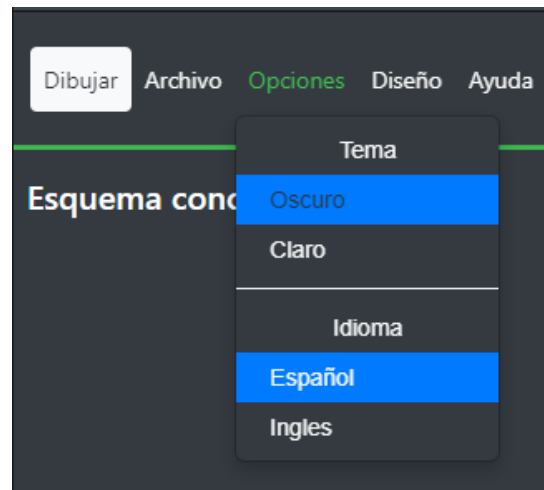
## 8.2.2 Diseño



Permite elegir el diseño en el que se dispone de información del diseño. Por defecto esta herramienta tiene el diseño más completo de todas las demás opciones de diseño, es decir, se divide en tres paneles, se describen a continuación comenzando de izquierda a derecha:

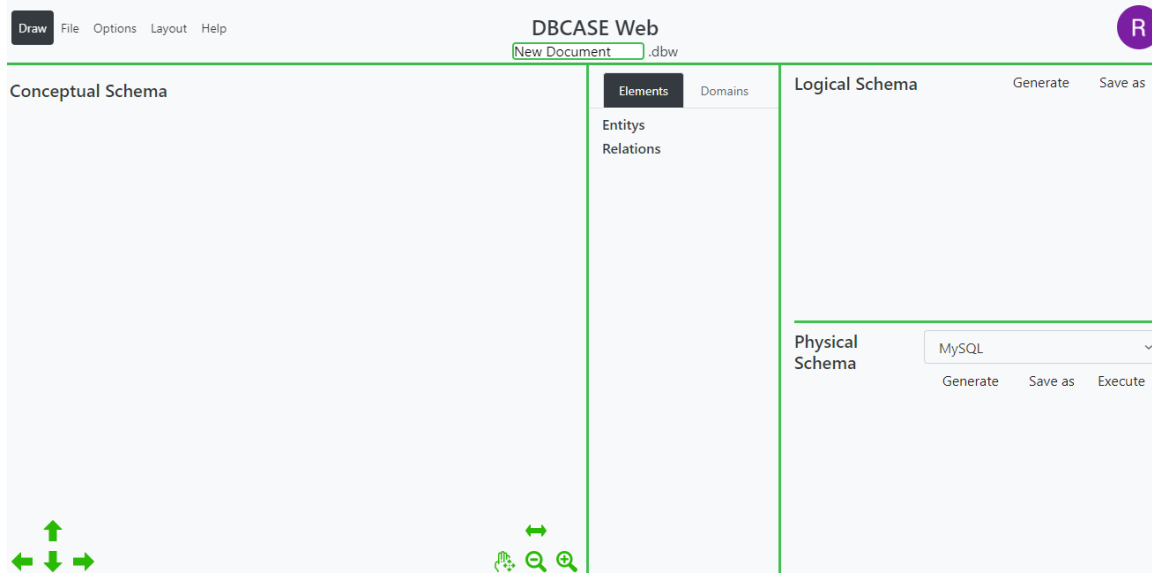
- Primer panel o EC ( Esquema conceptual), aquí es dónde se dibuja el diseño gráficamente.
- Segundo panel o E/D ( Elementos/Dominio) :
  - **Elementos:** se muestra un esquema en forma de árbol del modelo entidad relación existente en el panel de diseño, y en él se puede ver de forma esquemática todos los elementos y cómo están relacionados.
  - **Dominio:** informa al usuario de las opciones disponibles que tiene para asociar con los atributos creados, también permite crear nuevos tipos de dominio que el usuario considere necesarios.
- Tercer panel o EL y EF (esquema lógico y esquema físico): este tercer panel se divide en dos
  - EL o Esquema Lógico: panel donde se expresan las restricciones lógicas que se aplican a los datos pudiendo definir también las restricciones de integridad y restricciones perdidas.
  - EF o Esquema Físico: panel donde se dispone cómo se almacenan los datos físicamente en un sistema de almacenamiento en término de archivos e índices.

## 8.2.3 Opciones



Permite elegir opciones visuales tales como el idioma o el tema, por defecto el tema es *Oscuro* y el idioma es *Español*.

Si se eligiese tema *Claro* e idioma *Inglés*, queda de la siguiente manera

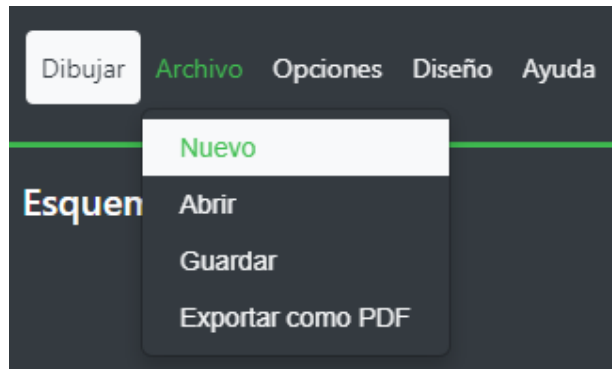


## 8.2.4 Archivo

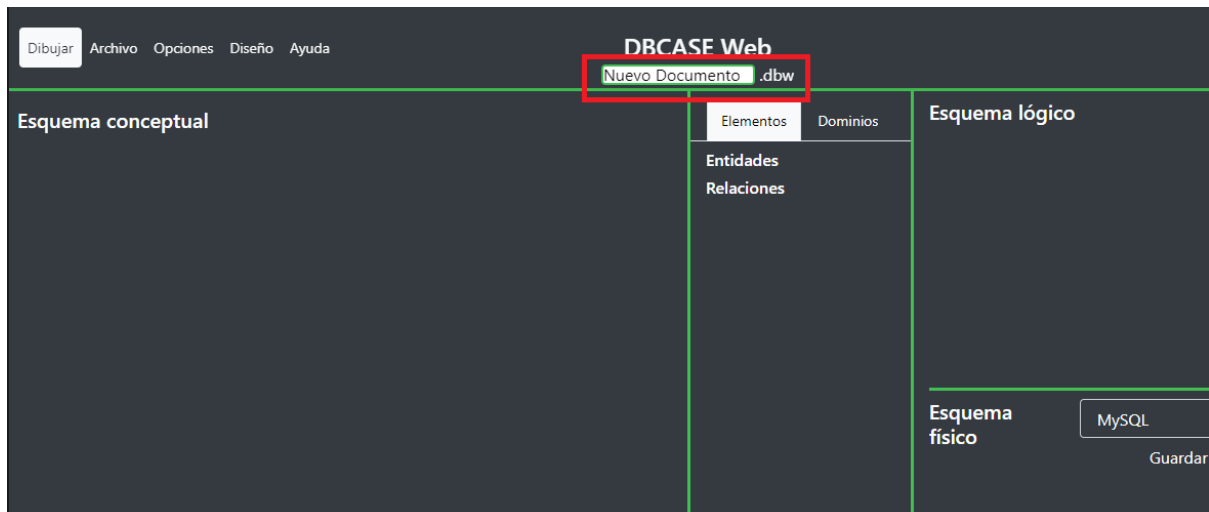
Dispone de las siguientes opciones



### 8.2.4.1 Nuevo

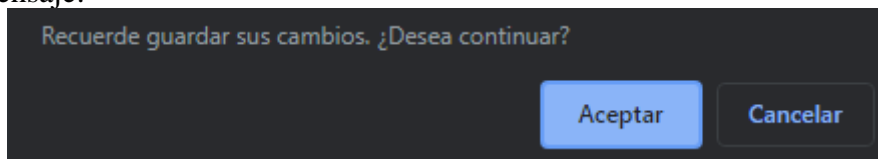


Limpia todos los paneles que estuviesen con información, es decir, deja el estado de la pantalla preparado para un nuevo diseño.



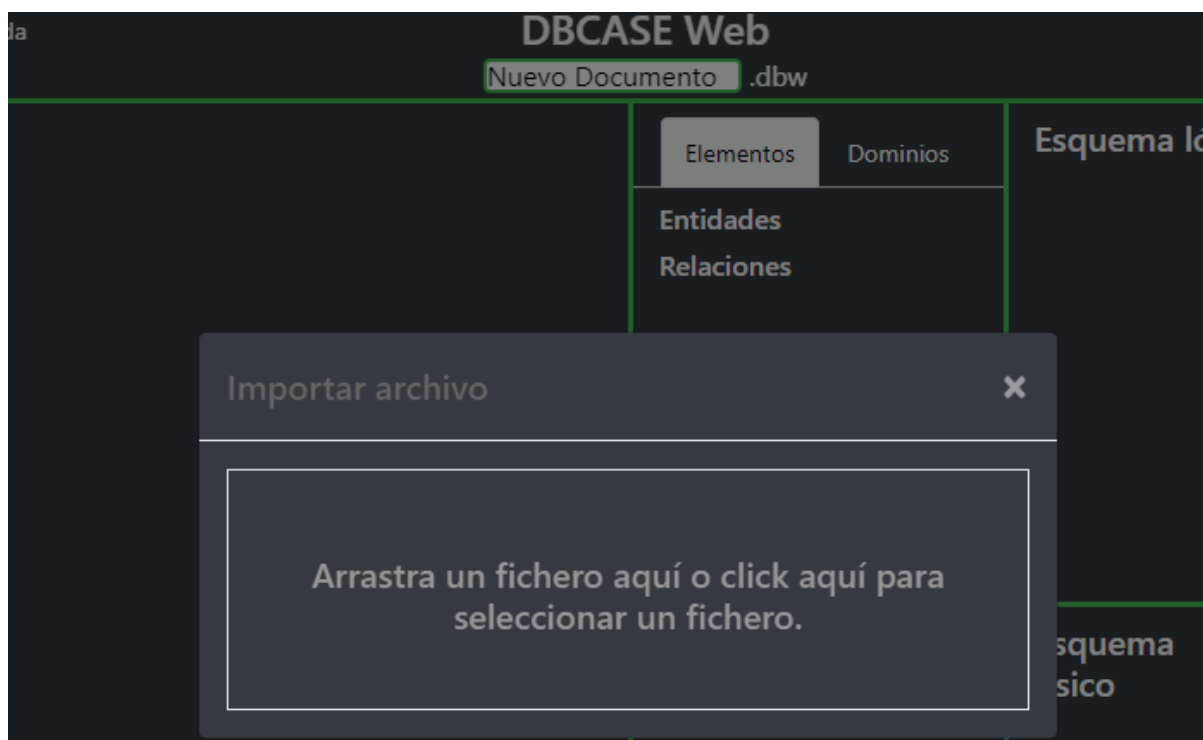
En el cuadro marcado en rojo, justo debajo del título DBCASE Web, aparecerá “*Nuevo Documento*” o en inglés “*New Document*”.

Si se tiene algo en alguno de los esquemas, al elegir esta opción el sistema muestra el siguiente mensaje:

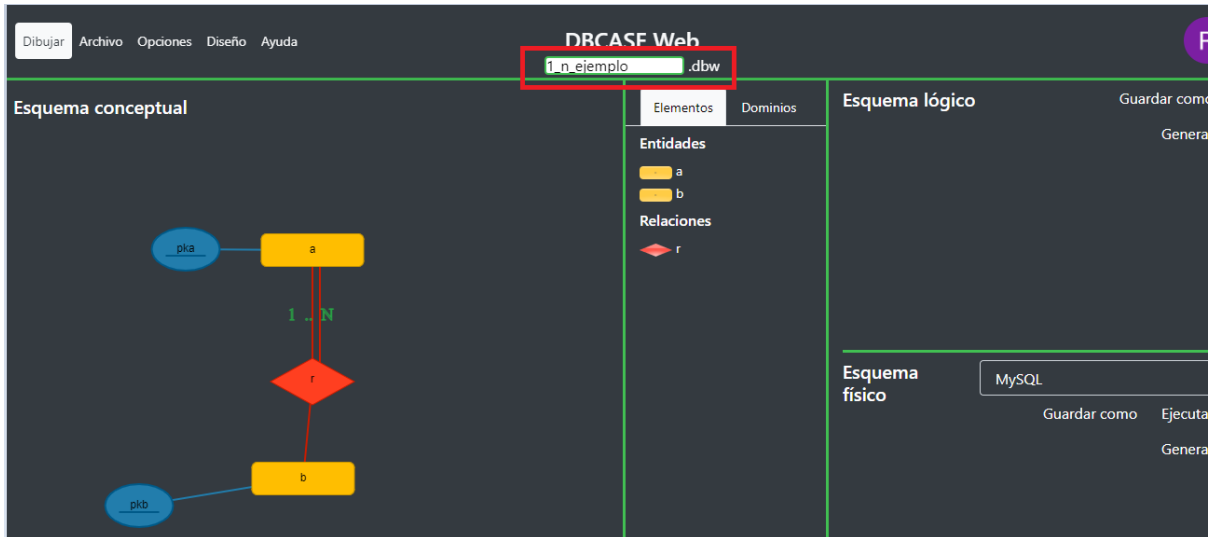


#### 8.2.4.2 Abrir

Permite abrir un diseño ya creado, para ello se abre un modal en el que se puede arrastrar el archivo .dbw o buscarlo en el explorador de archivo si se hace click en dicho modal.

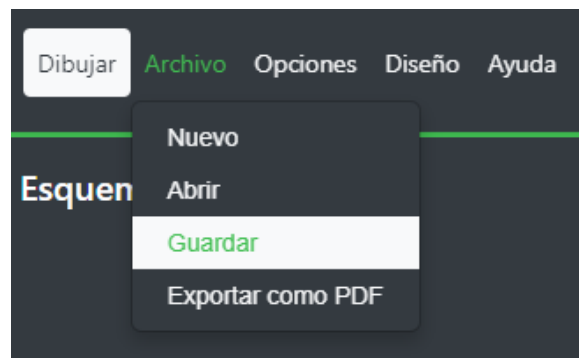


Al abrir un archivo se muestra de la siguiente manera:

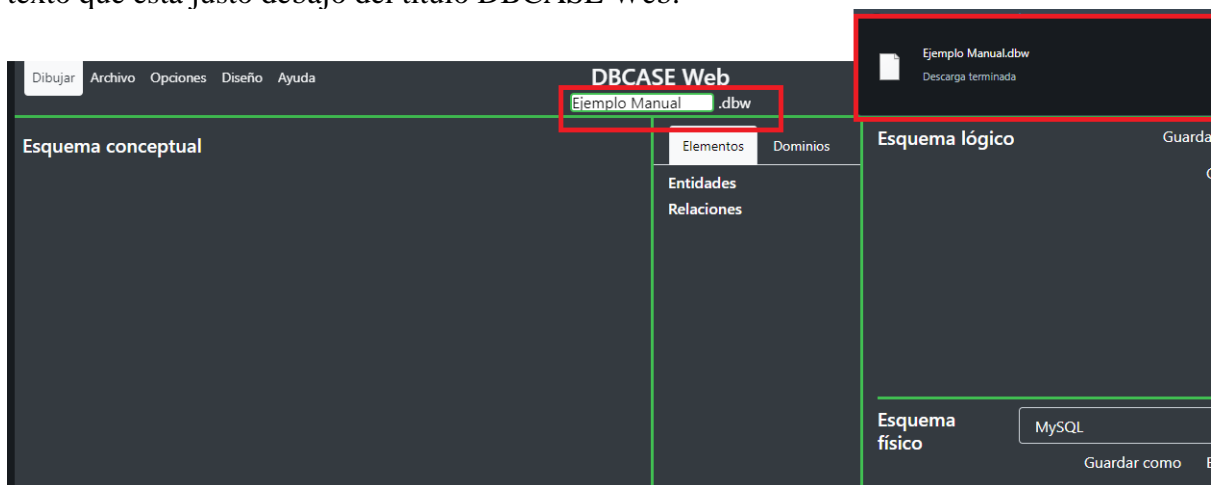


En el panel de Esquema conceptual, se dibuja el diseño y en el cuadro marcado con rojo justo debajo del título DBCASE Web, se puede ver el nombre del archivo que se ha abierto.

### 8.2.4.3 Guardar



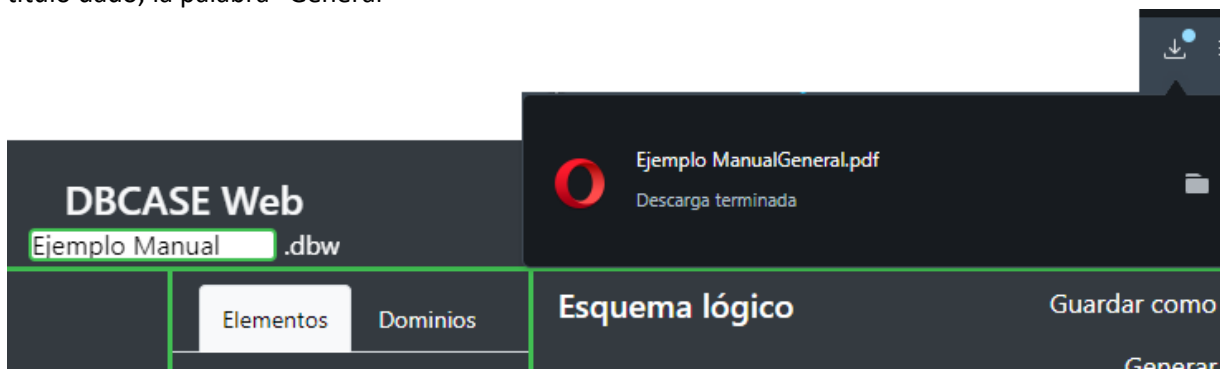
Se guarda el archivo con extensión .dbw con el nombre que tenga el documento del cuadro de texto que está justo debajo del título DBCASE Web.



#### 8.2.4.4 Exportar como pdf



Así como la opción guardar con esta opción se descarga el diseño con extensión .pdf, y se agrega al título dado, la palabra “General”



#### 8.2.5 Dibujar

Se despliega una lista de figuras definidas para realizar el diseño del esquema conceptual

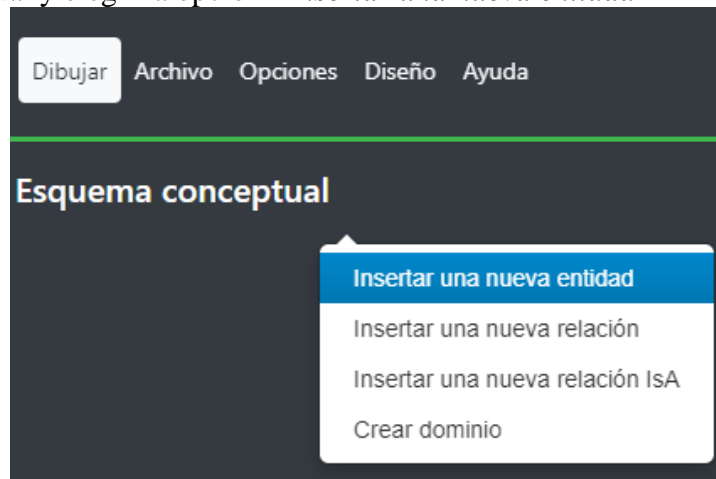


### 8.2.5.1 Entidad

Al seleccionar dibujar entidad, aparece un modal para definir el nombre y si es una entidad débil.

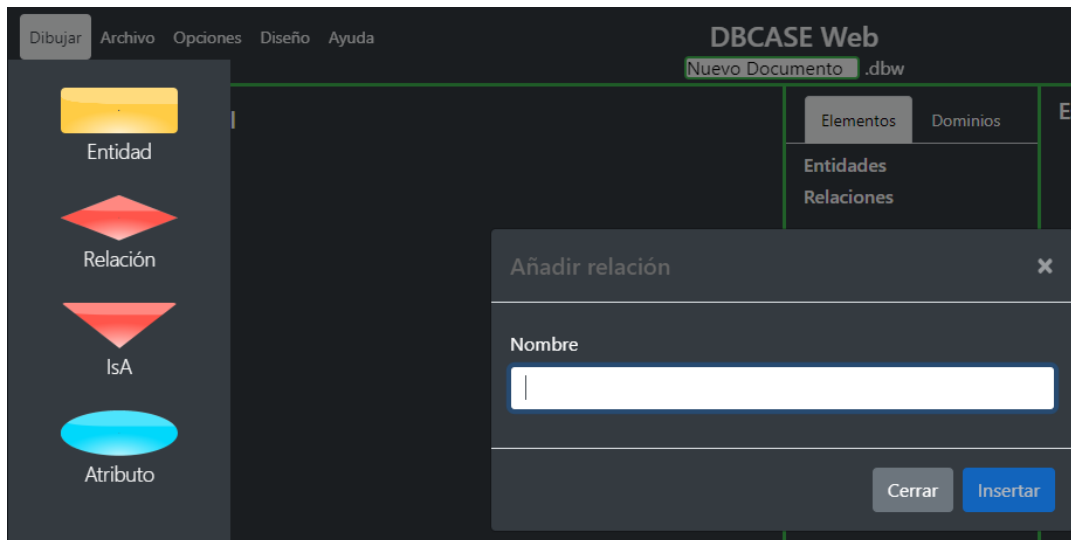
A dark-themed modal window titled "Añadir entidad" with a close button (X) in the top right corner. Below the title is a label "Nombre" followed by a white text input field. Underneath the input field is a checkbox labeled "Es una entidad débil.". At the bottom right of the modal are two buttons: "Cerrar" (grey) and "Insertar" (blue).

Para añadir una entidad también se puede hacer con el click derecho sobre el panel de Esquema conceptual y elegir la opción “Insertar una nueva entidad”

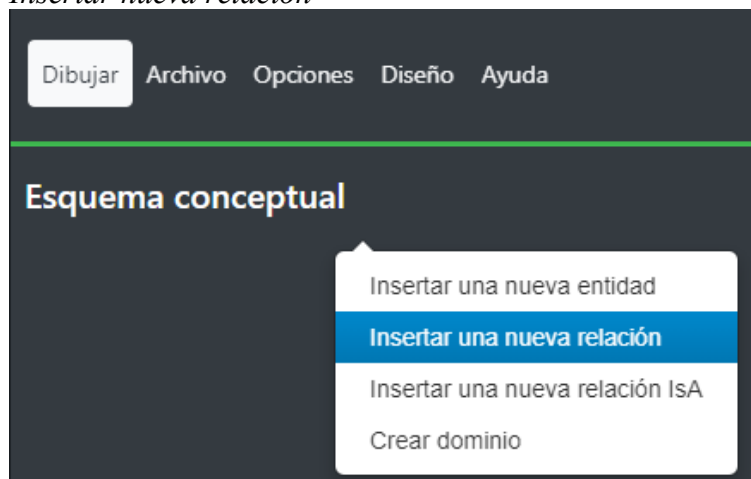


### 8.2.5.2 Relación

Al seleccionar dibujar una relación, aparece un modal que permite definir el nombre de la misma.



Se puede añadir una relación haciendo click derecho sobre el panel de Esquema conceptual y elegir la opción “*Insertar nueva relación*”



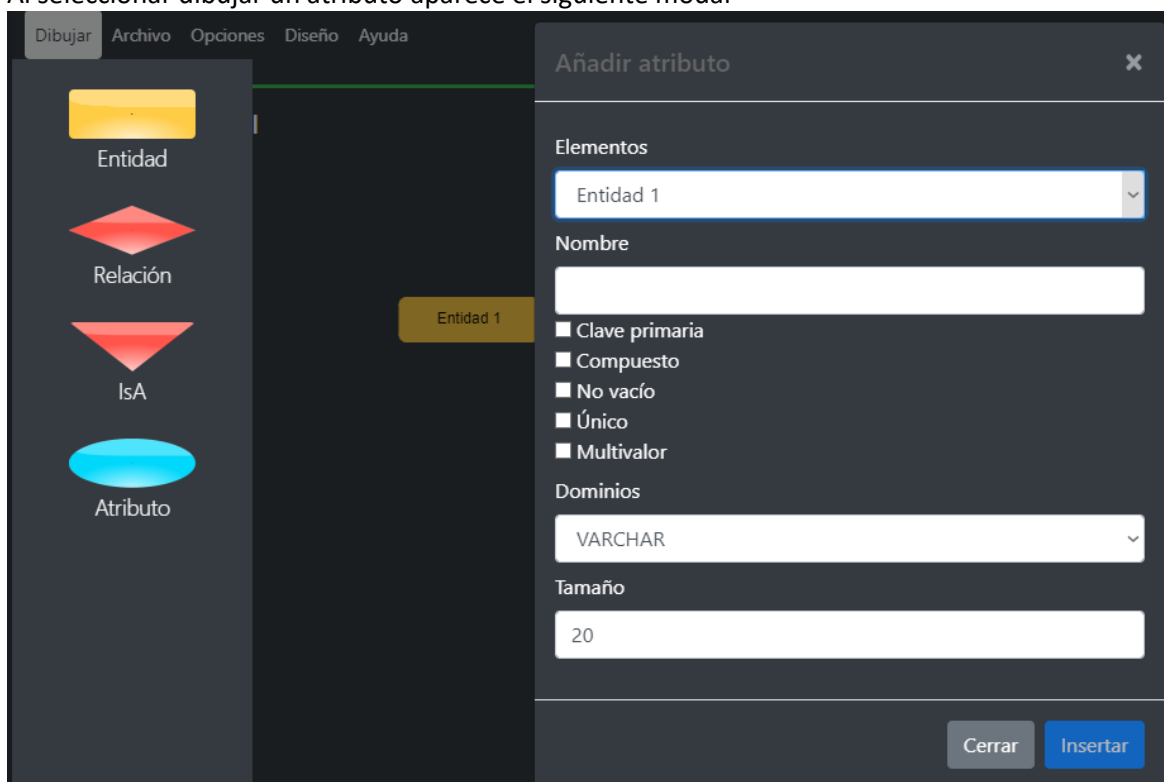
### 8.2.5.3 IsA

Al seleccionar dibujar IsA se dibuja automáticamente un elemento de este tipo, como sucede con entidad y relación, este elemento puede ser insertado al hacer click derecho sobre el panel Esquema conceptual.

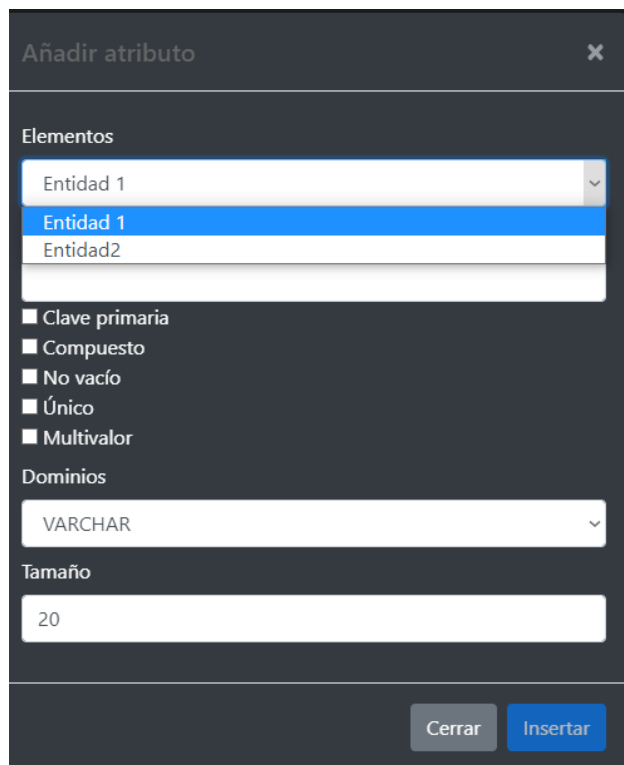


### 8.2.5.4 Atributo

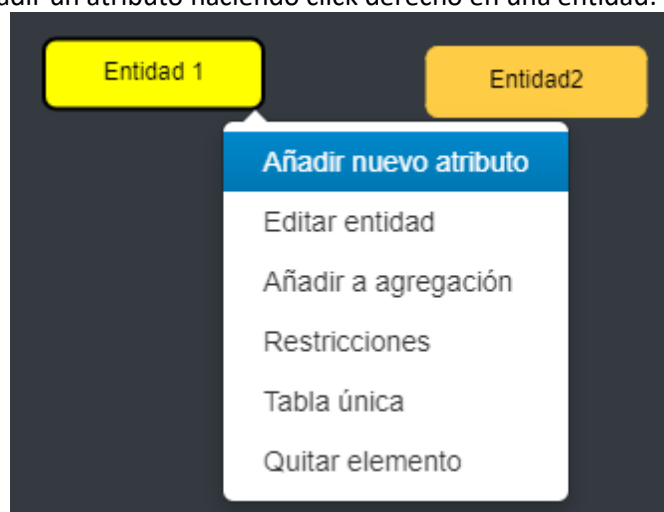
Al seleccionar dibujar un atributo aparece el siguiente modal



En el desplegable "Elementos", se puede elegir la entidad a la que se añade el atributo



Además, se puede elegir las restricciones lógicas que tiene dicho atributo, si es no vacío, multivalor, único, compuesto. Y su dominio junto con el tamaño de este. También se puede añadir un atributo haciendo click derecho en una entidad:

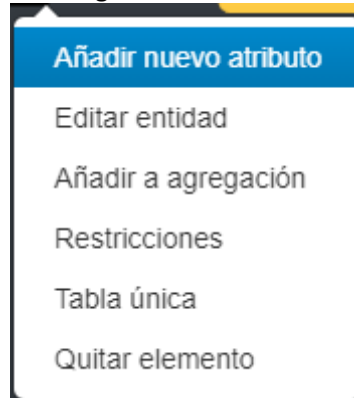


## 8.3 Funcionalidad sobre el esquema conceptual

Este apartado describe todas las funcionalidades sobre el esquema conceptual, centrándose en la acción de click derecho sobre algún elemento.

### 8.3.1 Entidad

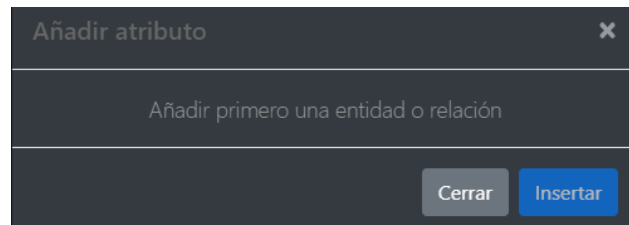
Sobre una entidad se pueden realizar las siguientes funciones :



### 8.3.1.1 Añadir nuevo atributo

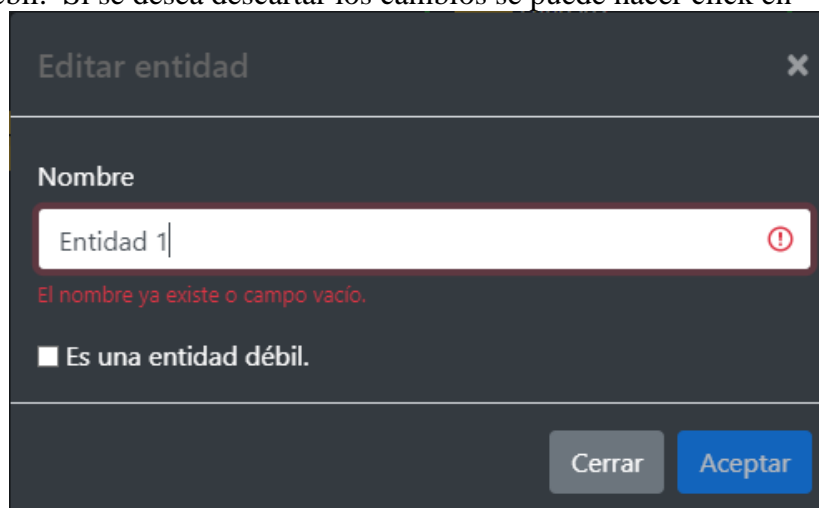
Descrito en la sección [4.2.5.4](#)

Además, si no existe entidad o relación a la que añadir un atributo, se muestra el siguiente modal



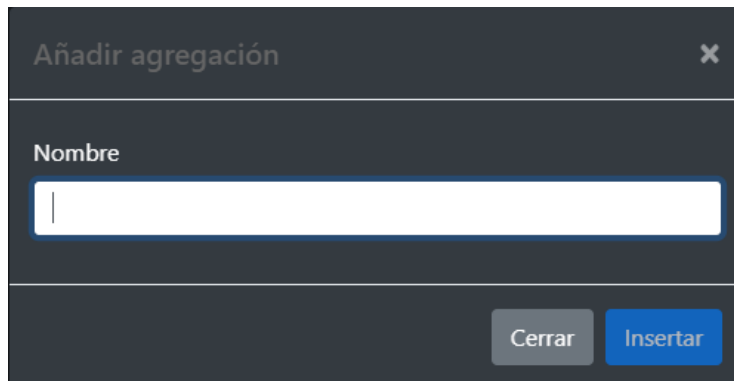
### 8.3.1.2 Editar entidad

Se abre un modal donde permite editar la entidad modificando el nombre y poder elegir si es una entidad débil. Si se desea descartar los cambios se puede hacer click en "Cerrar"



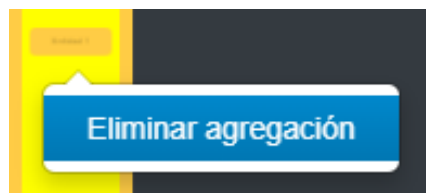
### 8.3.1.3 Añadir a agregación

Permite añadir dicha entidad a la agregación del nombre que se defina

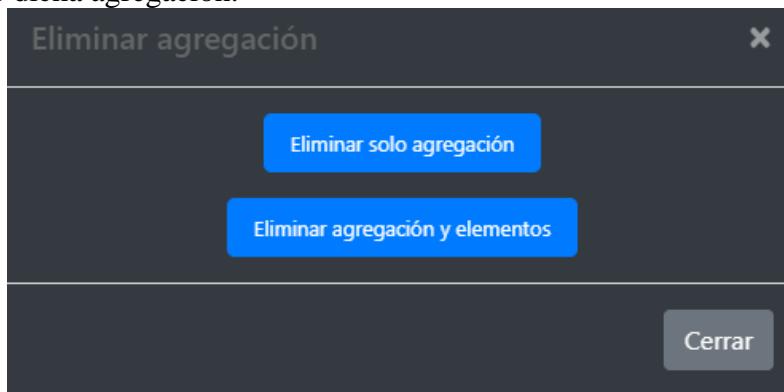


#### 8.3.1.3.1 Eliminar agregación

Al hacer click derecho en una agregación, aparece la siguiente opción:

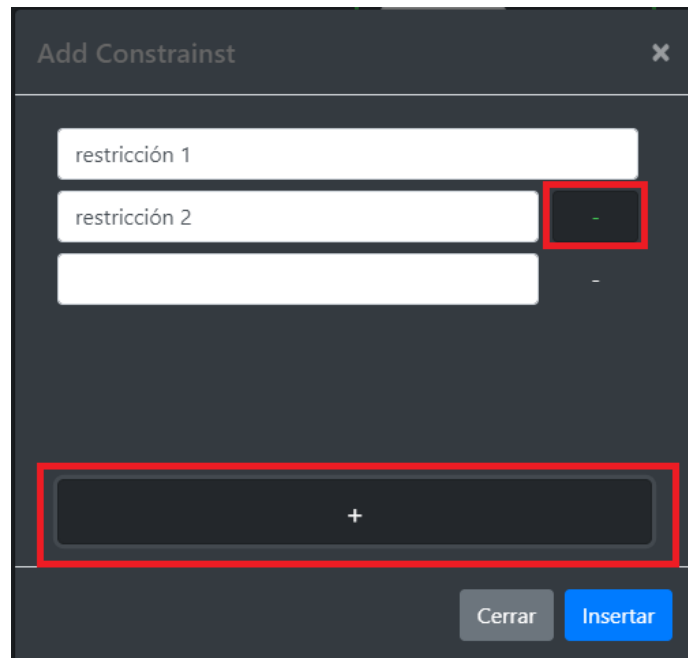


Si se elige esta opción se muestra el siguiente modal que permite eliminar sólo la agregación o lo que incluye dicha agregación.



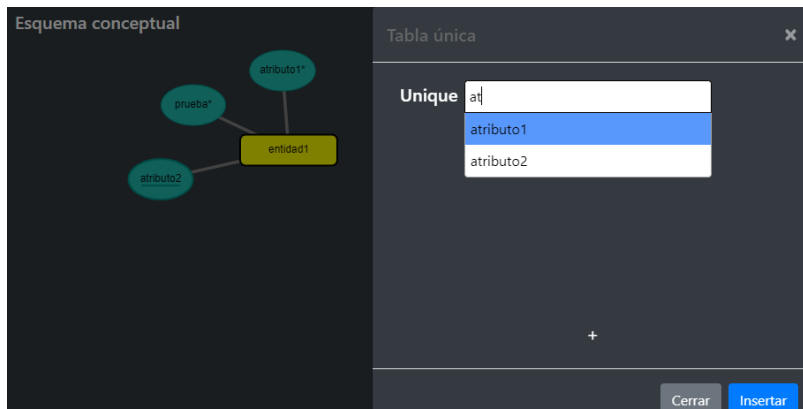
#### 8.3.1.4 Restricciones

Se muestra un modal que permite configurar todas las restricciones de dicha entidad. Hacer click en el "+" para añadir nueva restricción o hacer click en el "-" para eliminar la seleccionada (se indica resaltado en rojo para este ejemplo).

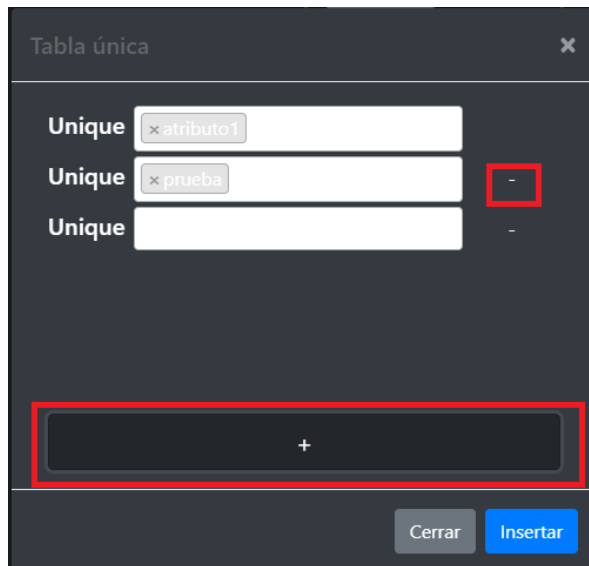


### 8.3.1.5 Tabla única

Se muestra un modal que permite definir con cual de los atributos se hace una tabla única. Se lista los atributos que pertenece a dicha entidad, aunque también se ofrece introducir por teclado el nombre y se filtran si coincide con el texto introducido.



En este modal también se ofrece definir varias tablas únicas.

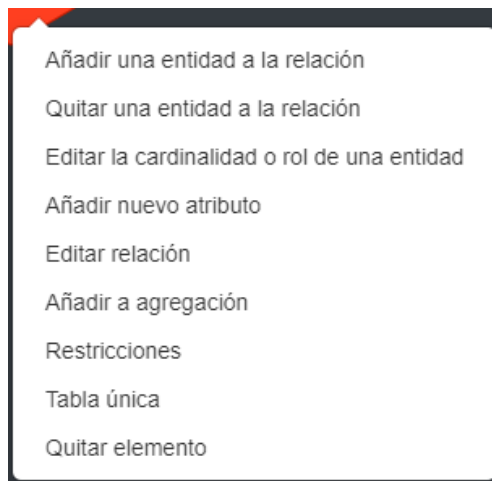


### 8.3.1.6 Quitar elemento

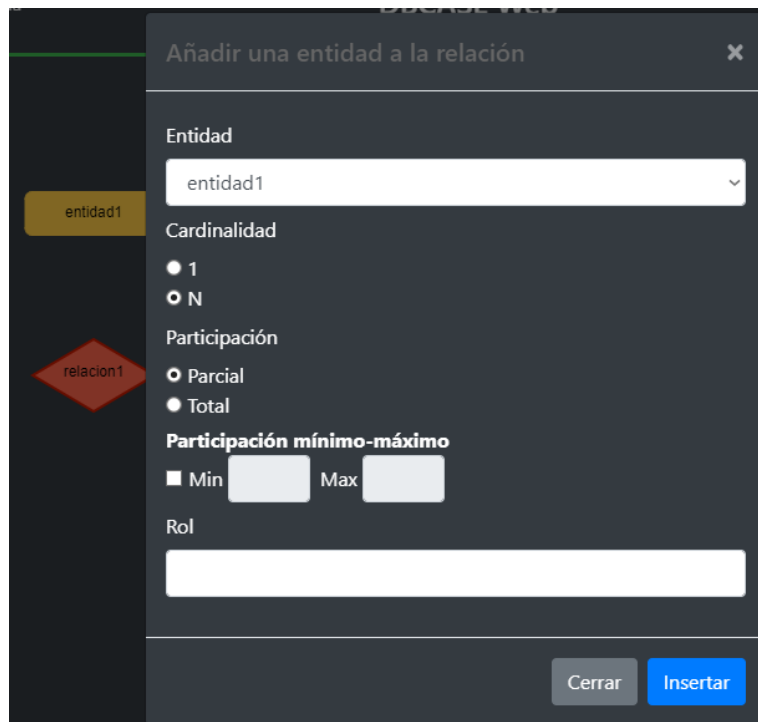
Eliminar tanto la entidad como sus atributos, esta acción puede realizarse con “supr” del teclado si la entidad es seleccionada.

### 8.3.2 Relación

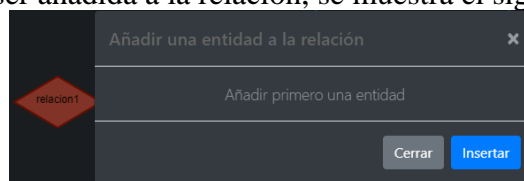
Sobre una relación se pueden realizar las siguientes acciones:



### 8.3.2.1 Añadir una entidad a la relación

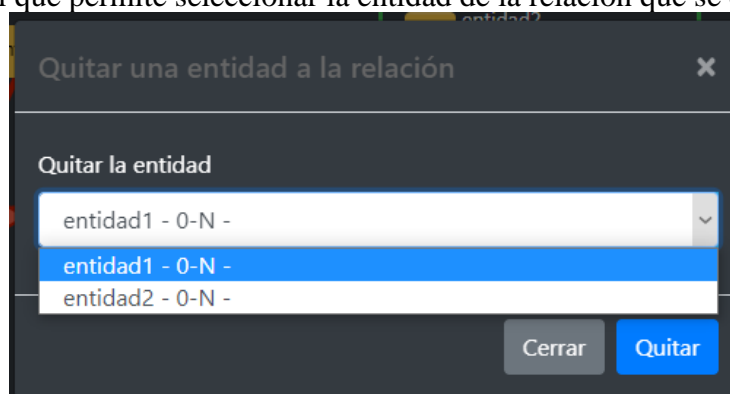


Permite añadir una entidad a la relación, así como indicar su cardinalidad, participación y rol. Si no existe entidad para ser añadida a la relación, se muestra el siguiente modal:



### 8.3.2.2 Quitar una entidad a la relación

Muestra un modal que permite seleccionar la entidad de la relación que se desea eliminar



### 8.3.2.3 Editar cardinalidad o rol de una entidad

Muestra un modal que permite editar la cardinalidad , participación y rol:

Editar la cardinalidad o rol de una entidad

Entidad  
entidad1

Rol  
entidad1 - 0-N -

Cardinalidad  
 1  
 N

Participación  
 Parcial  
 Total

Participación mínimo-máximo  
Min  Max

Rol

Cerrar Aceptar

### 8.3.2.4 Añadir nuevo atributo

Descrito en la sección [4.2.5.4](#)

### 8.3.2.5 Editar relación

Muestra un modal que permite modificar el nombre de dicha relación.

### 8.3.2.6 Añadir a agregación

Descrito en la sección [4.3.1.3](#)

### 8.3.2.7 Restricciones

Descrito en la sección [4.3.1.4](#)

### 8.3.2.8 Tabla única

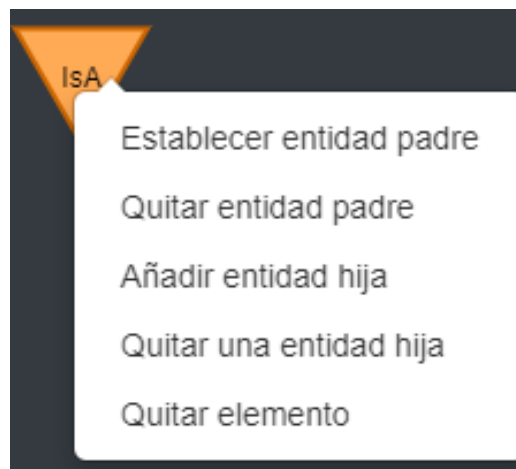
Descrito en la sección [4.3.1.5](#)

### 8.3.2.9 Quitar elemento

Elimina la relación pero no las entidades agregadas a dicha relación

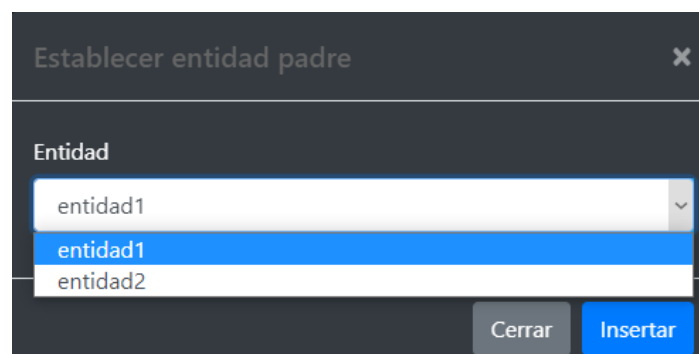
### 8.3.3 IsA

Sobre un elemento IsA se pueden realizar las siguientes acciones



#### 8.3.3.1 Establecer entidad padre

Muestra un modal que permite desplegar la lista de entidades del diseño y establecer la entidad padre de esta relación como se muestra a continuación:

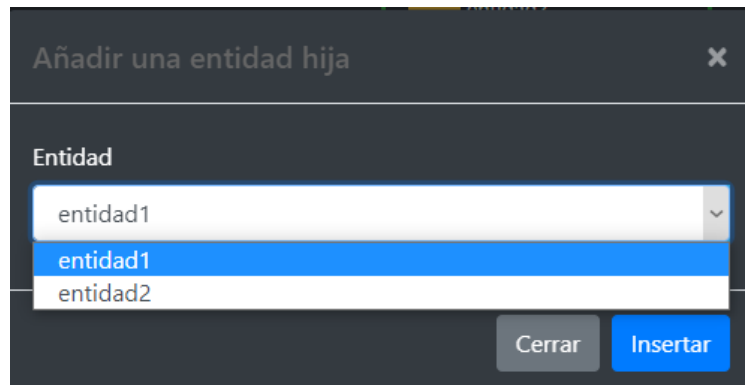


#### 8.3.3.2 Quitar entidad padre

Elimina la entidad padre que tenga asignada.

### 8.3.3.3 Añadir entidad hija

Muestra un modal que permite desplegar la lista de entidades candidatas de establecer como entidad hija



### 8.3.3.4 Quitar una entidad hija

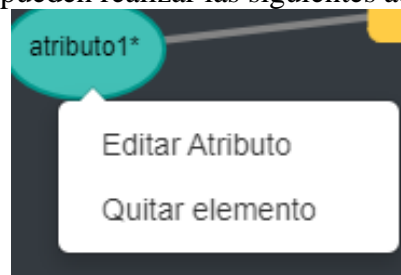
Muestra un modal que permite desplegar la lista de entidades hijas de la relación candidatas a ser eliminadas.

### 8.3.3.5 Quitar elemento

Elimina la relación IsA.

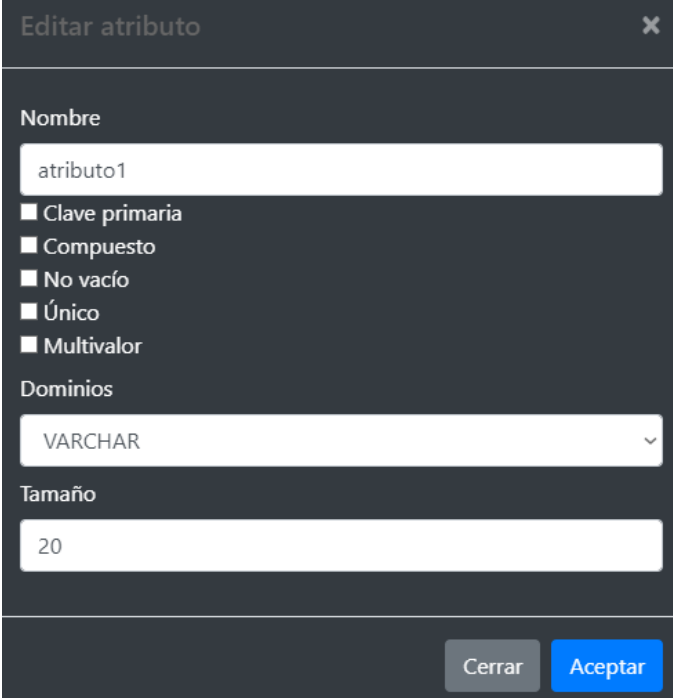
## 8.3.4 Atributo

Sobre un elemento atributo se pueden realizar las siguientes acciones:



### 8.3.4.1 Editar atributo

Muestra un modal que permite editar dicho atributo.



Editar atributo

Nombre

atributo1

Clave primaria

Compuesto

No vacío

Único

Multivalor

Dominios

VARCHAR

Tamaño

20

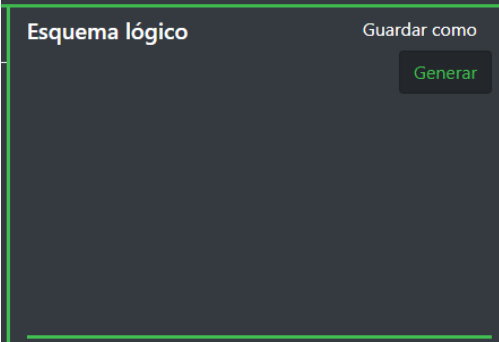
Cerrar Aceptar

### 8.3.4.2 Quitar elemento

Elimina dicho atributo, esta acción también puede ser realizada con la tecla “supr” del teclado.

## 8.4 Esquema lógico

Para poder generar un esquema lógico antes tiene que existir un esquema conceptual válido.



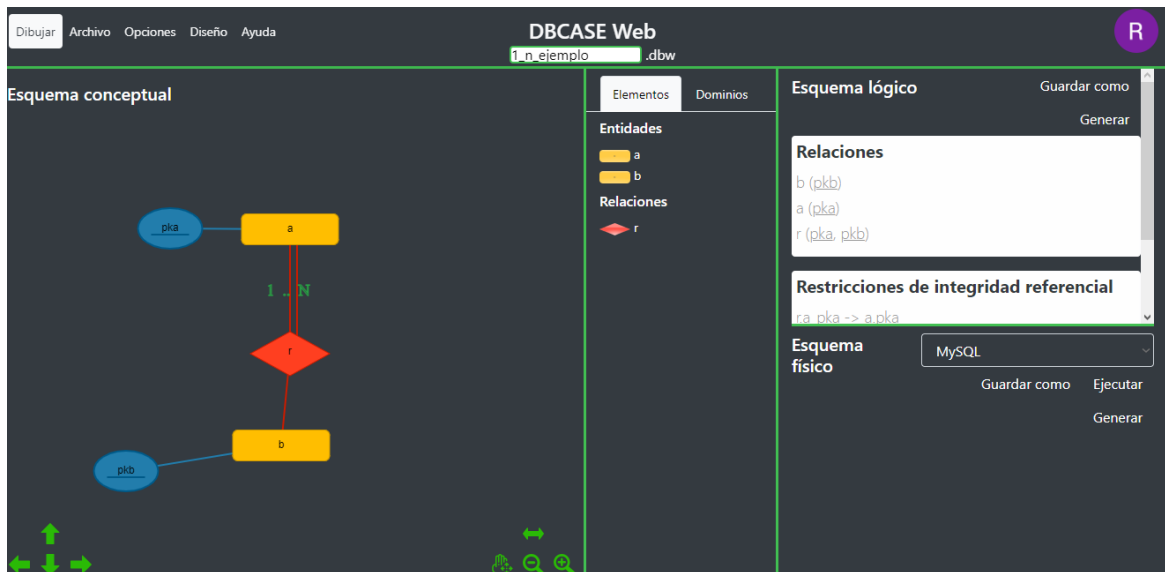
Esquema lógico

Guardar como

Generar

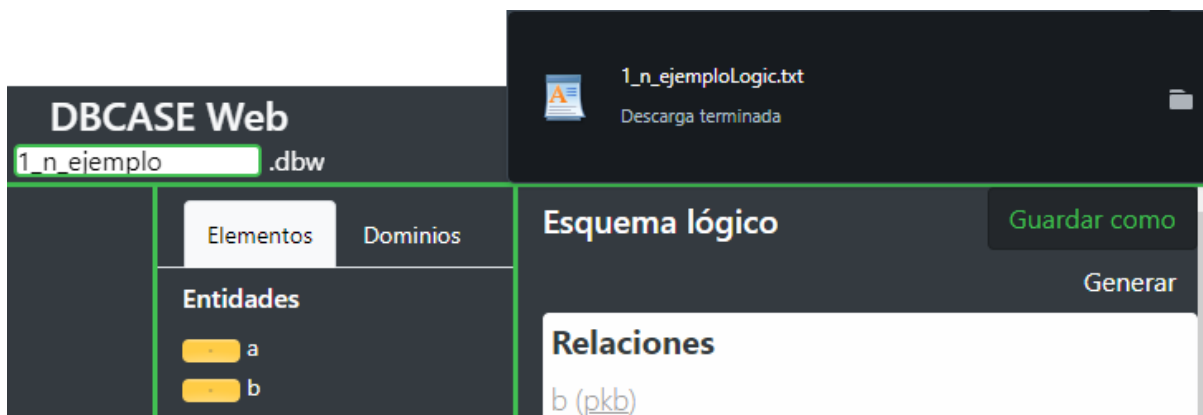
### 8.4.1 Generar

Genera el esquema lógico del esquema conceptual, y lo presenta en ese panel como se muestra a continuación



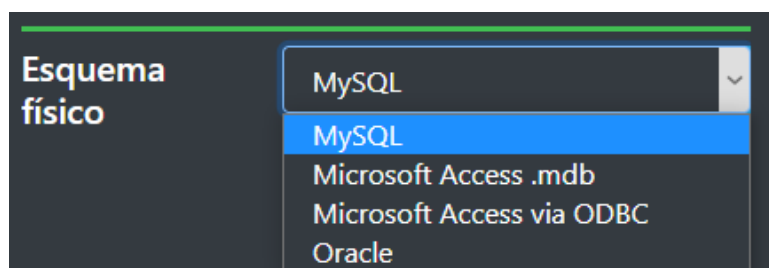
### 8.4.2 Guardar como

Para poder guardar el esquema lógico, antes tiene que haber sido generado. Se guarda el archivo agregando al nombre dado, la palabra “Logic” con extensión .txt



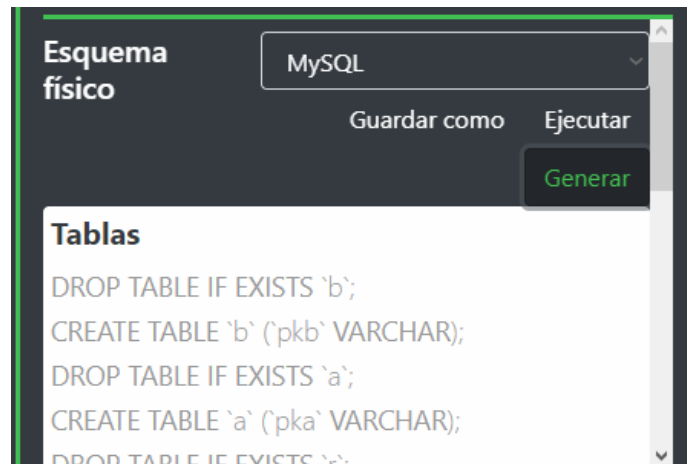
## 8.5 Esquema físico

Permite generar un esquema físico con los lenguajes que se muestran a continuación:



### 8.5.1 Generar

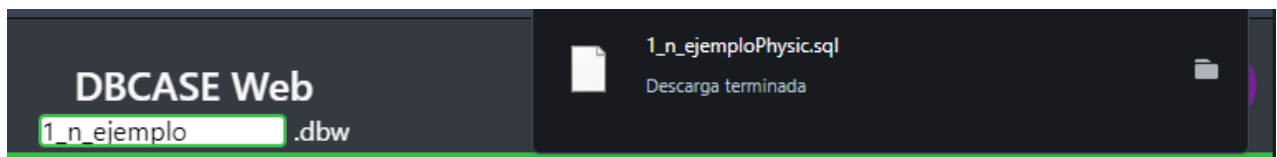
Así como en el esquema lógico, el esquema físico no puede ser generado sin tener un esquema conceptual válido. Al elegir generar, se muestra en el panel de esta manera:



### 8.5.2 Guardar como

Para poder guardar el esquema físico, tiene que haber sido generado.

Se guarda el archivo agregando al nombre dado, la palabra “Physic” con extensión .sql



## 8.6 Introducción al sistema

En esta sección se define el conocimiento necesario sobre versiones o estructuras de directorios que serán de gran ayuda para siguientes versiones.

El proyecto está hecho con Maven 4.0, que es un gestor y constructor de proyectos con la gran ventaja que la configuración de librerías y dependencias, se concentran en un único archivo, pom.xml(ubicado en el directorio raíz del proyecto). En este archivo se puede consultar y configurar la versión de java con la que se trabaja (jdk 1.8), las versiones de las librerías y dependencias y temas relacionados con configuración.

- **main/java** directorio donde se encuentra toda la lógica del programa, clases java.
- **main/resources** se encuentran los elementos estáticos (plantillas html, ficheros de configuración, css, imágenes, etc).
- **main/resources/static/js** se encuentran todos los ficheros JavaScript que utiliza el programa.
- **Target** almacena los .war generados de la versión.

### Generar archivo .war :

En cualquier IDE, se debe realizar, con Maven, *clean* y luego *install* desde las opciones Maven del IDE o también se puede ejecutar los comandos *mvn clean* y luego *mvn install*

**Ejecutar archivo .war** almacenado en el directorio target, se lanza el siguiente comando por consola:

```
java -jar DBCaseWeb-0.0.1-SNAPSHOT.war
```

y luego en el navegador se accede a la siguiente dirección:

<http://localhost:8080>