



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por Juan Romera Pérez
en Universidad de Burgos — 9 de julio de 2024
Tutor: Jesus Manuel Maudes Raedo

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	11
B.1. Introducción	11
B.2. Objetivos generales	11
B.3. Catálogo de requisitos	12
B.4. Especificación de requisitos	14
Apéndice C Especificación de diseño	43
C.1. Introducción	43
C.2. Diseño de datos	43
C.3. Diseño procedimental	46
Apéndice D Documentación técnica de programación	53
D.1. Introducción	53
D.2. Estructura de directorios	53
D.3. Manual del programador	54
D.4. Compilación, instalación y ejecución del proyecto	54

Apéndice E Documentación de usuario	59
E.1. Introducción	59
E.2. Requisitos de usuarios	59
E.3. Manual del usuario	59
Apéndice F Anexo de sostenibilización curricular	77
F.1. Introducción	77
F.2. Sostenibilidad ambiental	77
F.3. Sostenibilidad social	78
F.4. Sostenibilidad económica	78
F.5. Integración de la sostenibilidad en el currículum	78
F.6. Conclusiones	79
Bibliografía	81

Índice de figuras

B.1. Diagrama de Casos de Uso	15
C.1. Diagrama de clases	46
C.2. Diagrama secuencia añadir Tabla/Columna	48
C.3. Diagrama secuencia añadir relación	49
C.4. Diagrama secuencia actualizar relaciones	50
C.5. Diagrama secuencia cambio clave primaria	51
C.6. Diagrama secuencia mover columna	52
D.1. Clonación con Git GUI	55
D.2. Git GUI	56
D.3. Clonación Git Bash	56
D.4. Preview Vercel	57
D.5. Live Server	58
E.1. Aplicación	60
E.2. Icono tabla	61
E.3. Añadir tabla	61
E.4. Nueva tabla añadida	62
E.5. Icono columna	63
E.6. Añadir columna desde menú popup	64
E.7. Icono flecha	65
E.8. Soltar para añadir relación	65
E.9. Nueva relación creada	65
E.10. Panel propiedades	66
E.11. Apartado relleno	67
E.12. Apartado texto	68
E.13. Formulario para añadir unique compuesto	70

E.14. Formulario editar relación	71
E.15. Panel edición	72
E.16. Panel vista	73
E.17. Generación de código	74

Índice de tablas

A.1. Costes Hardware	7
A.2. Costes Software	7
A.3. Costes Personal	8
A.4. Costes Total	8
A.5. Licencias	9
B.1. CU-01 Gestión del grafo	16
B.2. CU-02 Importar diagrama	17
B.3. CU-03 Exportar diagrama	18
B.4. CU-04 Limpiar diagrama	19
B.5. CU-05 Mostrar diccionario de datos	20
B.6. CU-06 Gestión de tablas	21
B.7. CU-07 Añadir tabla	22
B.8. CU-08 Editar tabla	23
B.9. CU-09 Eliminar tabla	24
B.10.CU-10 Añadir Unique compuesto	25
B.11.CU-11 Eliminar Unique compuesto	26
B.12.CU-12 Gestión de columnas	27
B.13.CU-13 Añadir columna	28
B.14.CU-14 Editar columna	29
B.15.CU-15 Eliminar columna	30
B.16.CU-16 Mover posición	31
B.17.CU-17 Gestión de relaciones	32
B.18.CU-18 Añadir relación	33
B.19.CU-19 Editar relación	34
B.20.CU-20 Eliminar relación	35
B.21.CU-21 Gestión de claves	36
B.22.CU-22 Añadir clave foránea	37

B.23.CU-23 Eliminar clave foránea	38
B.24.CU-24 Generación de código	39
B.25.CU-25 Crear SQL	40
B.26.CU-25 Crear SQLAlchemy	41

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En el siguiente anexo se realizará un análisis de la planificación temporal junto con la viabilidad legal y económica del proyecto.

A.2. Planificación temporal

Introducción

Para la planificación temporal del proyecto se ha seguido la metodología *Scrum*, como se menciona en la memoria.

El proyecto se ha dividido en dos fases, estando comprendida la primera entre el Sprint 0 y el Sprint 2, siendo esta una fase enfocada a la formación en diferentes elementos del desarrollo web y a familiarizarse con la librería de Javascript *mxGraph*, fundamental en este proyecto. La segunda fase comprende entre el Sprint 3 y el Sprint Final, donde nos hemos centrado en el desarrollo e implementación de la aplicación y en la búsqueda de la información necesaria para llevar a cabo esta tarea.

Sprints

A continuación se detalla la planificación seguida durante los Sprints realizados, detallando los Issues realizados en cada uno.

Sprint 0

(05/03/2024 - 24/03/2024)

- **Formación en Javascript**
- **Formación en React**
- **Formación en Node**

Sprint 1

(25/03/2024 - 31/03/2024)

- **Formación en mxGraph.** Toma de contacto con la librería, estudiando y realizando diferentes ejemplos.
- **Crear la estructura de la documentación.** Creamos la estructura de la documentación en la plataforma *Overleaf*, siguiendo la plantilla indicada.

Sprint 2

(08/04/2024 - 14/04/2024)

- **Ajustar el tamaño de una celda a la longitud de su etiqueta.** Investigación en la librería mxGraph y Javascript para ajustar el tamaño de una celda al de su etiqueta.
- **Transformar las aristas en la notación de patas de cuervo.** Investigar como transformar los símbolos utilizados en la librería mxGraph al final de las aristas para representar la notación de patas de cuervo.

Sprint 3

(15/04/2024 - 28/04/2024)

- **Edición de columnas.** Estudiar como implementar añadir, eliminar y modificar columnas.
- **Diccionario de datos.** Implementación de un diccionario de datos, mostrando la información adicional de las columnas como tooltip.

- **Atributos de la tabla al crear una relación.** Estudiar que elementos se deben añadir a las tablas cuando se crea una nueva relación entre dos tablas.
- **Ajustar el tamaño de tablas y columnas.** Ajustar el tamaño de tablas y columnas cuando alguno de los elementos que forman la tabla cambie, manteniendo una anchura y altura mínimos y máximos. Este Issue se descartó por problemas de la librería.

Sprint 4

(05/05/2024 - 12/05/2024)

- **Demo.** Implementación de una primera demostración con los conceptos estudiados hasta el momento.
- **Ajuste de la etiqueta mostrada al tamaño de la celda.** Ajustaremos la etiqueta que se muestra para las tablas y columnas al tamaño de estas. Teniendo en cuenta el número de caracteres que forman la etiqueta junto a la anchura de la celda, se mostrará completa o se cortará terminando en 3 puntos suspensivos (...), indicando que la etiqueta continua.

Sprint 5

(13/05/2024 - 26/05/2024)

- **Documentación.** Continuar con el desarrollo de la memoria.
- **Mejorar generación SQL.** Una vez se ha implementado correctamente la notación de patas de cuervo, podemos mejorar la generación de código SQL a partir del grafo diseñado por el usuario.
- **Error tooltip enlaces.** Corregir error que causa que la tooltip de los enlaces no se muestre.
- **Manejo de claves.** Hasta ahora solo se han tenido en cuenta claves primarias simples. Implementar manejo de claves foráneas y claves compuestas.
- **Refactorizar actualizarClaves.** La función actualizarClaves cuenta con numeroso código repetido que sería conveniente refactorizar

Sprint 6

(27/05/2024 - 09/06/2024)

- **Div con las propiedades.** Implementar un div lateral donde se muestren las propiedades de cada elemento, eliminando la ventana utilizada hasta ahora. El div implementado cambiará en función del elemento seleccionado, escondiéndose en caso de que no haya nada seleccionado.
- **Modificar tamaño y disposición de los div.** Modificar la disposición de los elementos de la interfaz de nuestra aplicación.
- **Nuevos iconos para tabla y columna.** Implementar nuevos iconos que identifiquen mejor las tablas y las columnas, más acordes con la nueva interfaz.
- **Problema relaciones 1:1.** Corregir problema en las relaciones 1:1 cuando uno de los lados es opcional. El símbolo se debe colocar en la tabla correspondiente y la clave foránea se debe situar en el lado opcional.
- **Mover los estilos a un fichero CSS.** Mover los estilos de los div y demás elementos de la interfaz a un fichero CSS para tener una mejor organización de los contenidos.

Sprint 7

(19/06/2024 - 25/06/2024)

- **Despliegue del proyecto.** Reorganizar el repositorio para una primera release. Una vez reorganizado, desplegar la aplicación utilizando la plataforma Vercel.
- **Integrar herramientas de CI.** Implementar en el repositorio herramientas de integración continua.
- **Implementar limpiar diagrama.** Implementar un botón en la aplicación que al pulsarlo elimine todos los elementos del diagrama introducidos por el usuario.
- **Añadir claves foráneas en relaciones ya creadas.** Cuando se añada una nueva clave primaria en una tabla, añadir esta columna como clave foránea donde sea necesario. La actualización de las claves

foráneas ocurre tanto cuando una columna pasa a ser clave primaria como cuando esta deja de serlo.

- **Refactorizar funciones.** Con SonarCloud se ha descubierto que las funciones *getLabel*, *actualizarClaves*, *createSql* y *configurarTabEstilos* tienen una complejidad cognitiva demasiado elevada. Es conveniente refactorizar estas funciones para un mantenimiento de la aplicación más sencillo. **Memoria.** Continuar con el desarrollo de la documentación. **Limpiar código.** Eliminar líneas de código innecesarias y mejoras de facilidad de mantenimiento de código.

Sprint 8

(26/06/2024 - 02/07/2024)

- **Unique compuesto.** Implementar una opción en las tablas que permita establecer Unique compuestos. Estos Unique compuestos se almacenarán en las tablas y se mostrarán en el código generado.
- **Ventana diccionario de datos.** Implementar una ventana donde se muestre el diccionario de datos de todas las columnas del diagrama. Se listarán las columnas de cada tabla una a una mostrando nombre, tipo, título y descripción.
- **Generar SQLAlchemy.** Implementar la generación de código para SQLAlchemy a partir de diagrama del usuario.
- **Propiedades claves foráneas.** Corregir las propiedades NOT NULL y UNIQUE de las claves foráneas. Estas propiedades se establezcan automáticamente según el tipo de relación que dispongamos.
- **Detección de errores importar XML.** Implementar detección de errores a la hora de importar un diagrama desde un fichero XML. En caso de que el fichero contenga errores, no se importará nada.
- **Modificar bucles for por bucles for of.** Simplificar bucles donde sea necesario utilizando for of en vez de for.
- **Implementar control por teclado.** Configurar el diagrama para utilizar atajos de teclado.
- **Asociar PK a FK.** Almacenar el id de la clave primaria asociada con la clave foránea para posterior acceso a sus datos. utilizar este id a la hora de mostrar el nombre al crear el código SQL y SQLAlchemy.

- **Extraer configuración estilos a XML.** Extraer la configuración de los estilos de los elementos del diagrama a un fichero XML desde el que importarlos.

Sprint 9

(02/07/2024 - 09/07/2024)

- **Reemplazar innerHTML por textContent**
- **Limpiar código y documentar.** Eliminar líneas de código innecesarias y añadir comentarios JSDoc.
- **Finalizar documentación**
- **Selección estilo línea.** Implementar una selección que permita al usuario elegir el estilo de línea de los enlaces entre línea continua y línea discontinua.

A.3. Estudio de viabilidad

Viabilidad económica

En el caso de que el proyecto se hubiera realizado con un enfoque empresarial y no académico, los costes y beneficios de la explotación del proyecto serían los que se detallan a continuación.

Costes

Esta sección detalla los distintos costes del proyecto.

Costes de Hardware

Los costes necesarios relativos al hardware para realizar este proyecto son únicamente un ordenador, portátil o de sobremesa, y sus periféricos. En este caso, se ha utilizado un portátil junto con unos periféricos adquiridos hace 6 años, asumiendo que la vida útil de un portátil entorno a los 1000 euros se encuentra entre los 4 y 7 años[2], y teniendo en cuenta que este material se ha utilizado para la realización de ejercicios anteriores en cursos pasados, podemos asumir la total amortización del equipo. Aun así, calculando con los 6 años utilizados, teniendo el equipo una vida útil de 7 años, obtenemos los siguientes costes.

Concepto	Coste(€)	Coste amortizado(€)
Portátil y periféricos	1000	857.14
Total	1000	857.14

Tabla A.1: Costes Hardware

Costes de Software

En este apartado se detallan los costes relativos al software, por lo que también se tendrá en cuenta costes de despliegue, almacenamiento y suscripciones de análisis de código e integración continua. Teniendo en cuenta que el resto de herramientas software empleadas son de uso gratuito, los costes son los siguientes.

Concepto	Coste(€)	Coste amortizado(€)
Vercel Pro	240	80
Codacy Pro	180	60
Sonarcloud Developer	160	53.33
Github Enterprise	252	84
Total	832	277.33

Tabla A.2: Costes Software

Estos costes se han calculado suponiendo un proyecto realizado en un entorno empresarial y asumiendo un dinero invertido durante 12 meses (1 año) de suscripciones, habiendo sido usadas durante 4 meses. Por simplicidad, los precios encontrados en dólares se han trasladado 1:1 a euros.

Coste de personal En este apartado se detalla el coste de tener a una persona desarrollando el proyecto durante 4 meses. Considerenado un salario mensual bruto de 2000 euros mensuales (12 pagas), los costes obtenidos son los siguientes.

Concepto	Coste(€)
Salario mensual bruto	2000
Retención IRPF	544.45
Seguridad Social	1085.30
Salario mensual que paga la empresa	3629.76
Total	14518.96

Tabla A.3: Costes Personal

Los costes se han calculado teniendo en cuenta un 15 % de IRPF [3] y un 29.9 % de cotización a la Seguridad Social [4].

Los cálculos del gasto que supone un trabajador a la empresa se obtienen calculando

$$\frac{2000}{1 - (0,299 + 0,15)} = 3629,76$$

Costes totales

Concepto	Coste
Hardware	857.14
Software	277.33
Personal	14518.96
Total	15653.23

Tabla A.4: Costes Total

Beneficios

La aplicación desarrollada se distribuirá de forma gratuita y no presenta ningún sistema de monetización implementado. La aplicación estará disponible de forma libre para todo el mundo, siendo una aplicación de código abierto mantenida por la comunidad.

Viabilidad legal

En este apartado se detallarán las licencias software de las dependencias del proyecto.

En la siguiente tabla se listan las licencias de las dependencias usadas.

Herramienta	Versión	Licencia
MxGraph	4.2.2	Apache 2.0
webpack	2.2.1	MIT

Tabla A.5: Licencias

La licencia utilizada por la librería *mxGraph* es Apache 2.0¹. Esta es la licencia menos permisiva que encontramos, por lo que nuestra licencia debe cumplir sus restricciones.

Para cumplir con las restricciones de la licencia Apache 2.0 debemos:

- Incluir una copia de la licencia Apache 2.0 en nuestro proyecto.
- Debemos avisar de cualquier modificación en alguno de los ficheros de la dependencia.
- Debemos conservar todos los avisos de derechos de autor, patente, marca registrada y atribución.
- Si la dependencia incluye algún fichero NOTICE debemos incluirlo también.
- No podemos trabajar o integrar la librería con un producto o servicio propiedad o comercializado por una entidad de *Atlassian*.

Cumpliendo con lo anterior, una de las licencias que mejor nos puede encajar es la licencia *MIT*. Esta es una licencia de código libre muy permisiva y compatible con la licencia Apache 2.0, lo que nos permitirá distribuir nuestra aplicación de forma libre y mantener el código fuente como código libre para que la comunidad pueda contribuir.

¹En el repositorio no aparece explícitamente Apache 2.0, pero lo podemos encontrar en la [la página de su paquete npm](#).

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este anexo se listarán los requisitos que definen el compartamiento de la aplicación desarrollada. Con esta lista de requisitos, se pretende dejar un escrito que sirva como documento contractual con el cliente y como documentación correspondiente a la aplicación. Se separa el contenido de este anexo en dos apartados:

- **Catálogo de requisitos:** requisitos funcionales y no funcionales del proyecto.
- **Casos de uso:** Explicación de forma más detallada de las funcionalidades de la aplicación.

B.2. Objetivos generales

Los objetivos generales del proyecto son:

- Desarrollar una aplicación web que permita el diseño de diagramas relacional con la notación de patas de cuervo.
- Almacenar los diagramas en un fichero con formato XML, para acceder a ellos posteriormente.
- Generar el código correspondiente al diagrama creado para su posterior implementación.

- Facilitar la creación de relaciones entre las tablas mediante la gestión automática de sus claves foráneas.

B.3. Catálogo de requisitos

A continuación se enumerarán los requisitos.

Requisitos funcionales

- **RF-1 Gestión del grafo:** La aplicación debe ser capaz de gestionar el grafo que contiene los elementos del diagrama relacional.
 - **RF-1.1 Importar diagrama:** La aplicación podrá crear un diagrama a partir de un fichero XML formateado correctamente.
 - **RF-1.2 Exportar diagrama:** La aplicación proveerá al usuario un método para exportar el diagrama en un fichero XML.
 - **RF-1.3 Limpiar diagrama:** El usuario podrá eliminar todos los elementos que aparecen en el diagrama.
 - **RF-1.4 Mostrar diccionario de datos:** El usuario podrá acceder a un diccionario de datos que muestre información sobre las columnas de cada tabla.
- **RF-2 Gestión de tablas:** La aplicación tiene que ser capaz de gestionar las tablas, que representan las entidades, introducidas en el diagrama.
 - **RF-2.1 Añadir tabla:** El usuario podrá añadir una nueva tabla al diagrama.
 - **RF-2.2 Editar tabla:** El usuario podrá editar los datos y el aspecto de las tablas introducidas en el diagrama.
 - **RF-2.3 Eliminar tabla:** El usuario podrá eliminar las tablas introducidas en el diagrama.
 - **RF-2.4 Añadir Unique compuesto:** El usuario podrá crear Uniques compuestos que se almacenarán en la tabla.
 - **RF-2.5 Eliminar Unique compuesto:** El usuario podrá eliminar los Uniques compuestos que haya añadido con anterioridad.
- **RF-3 Gestión de columnas:** La aplicación debe ser capaz de gestionar las columnas, representantes de los atributos de las entidades, introducidas en el diagrama.

- **RF-3.1 Añadir columna:** El usuario podrá añadir los atributos que necesite a cada entidad, representados como columnas.
- **RF-3.2 Editar columna:** El usuario podrá editar la información y el aspecto de las columnas.
- **RF-3.3 Eliminar columna:** El usuario podrá eliminar las columnas añadidas a una tabla.
- **RF-3.4 Mover posición:** El usuario podrá mover las columnas hacia arriba o hacia abajo dentro de una tabla.
- **RF-4 Gestión de relaciones:** La aplicación debe ser capaz de gestionar las relaciones entre las diferentes tablas del diagrama.
 - **RF-4.1 Añadir relación:** El usuario podrá añadir una relación entre dos tablas, conectándolas.
 - **RF-4.2 Editar relación:** El usuario podrá editar la información y el aspecto de las relaciones, siempre dentro de la notación de patas de cuervo.
 - **RF-4.3 Eliminar relación:** El usuario podrá eliminar una relación entre dos tablas.
- **RF-5 Gestión de claves:** La aplicación debe ser capaz de gestionar las claves primarias y foráneas de cada tabla.
 - **RF-5.1 Añadir clave foránea:** El sistema añadirá automáticamente las claves foráneas en las tablas que corresponda.
 - **RF-5.2 Eliminar clave foránea:** El sistema eliminará automáticamente las claves foráneas en las tablas que corresponda.
- **RF-6 Generación de código:** El sistema generará código a partir de los diagramas de usuario para diferentes lenguajes.
 - **RF-6.1 Crear SQL:** La aplicación proporcionará al usuario el código SQL correspondiente al diagrama relacional que se muestra en el grafo.
 - **RF-6.2 Crear SQLAlchemy:** La aplicación proporcionará al usuario la creación de las tablas correspondientes al diagrama para utilizar con SQLAlchemy.

Requisitos no funcionales

Los requisitos no funcionales son los mencionados a continuación[6].

- **RNF-1 Seguridad:** El sistema debe estar protegido frente al acceso no autorizado.
- **RNF-2 Actuación:** El sistema debe poder manejar el número requerido de usuario sin ninguna degradación en el rendimiento.
- **RNF-3 Escalabilidad:** El sistema debe ser capaz de escalar hacia arriba o hacia abajo según sea necesario.
- **RNF-4 Disponibilidad:** El sistema debe estar disponible cuando sea necesario.
- **RNF-5 Mantenimiento:** El sistema debe ser fácil de mantener y actualizar.
- **RNF-6 Portabilidad:** El sistema debe poder ejecutarse en diferentes plataformas con cambios mínimos.
- **RNF-7 Fiabilidad:** El sistema debe ser confiable y cumplir con los requisitos de usuario.
- **RNF-8 Usabilidad:** El sistema debe ser fácil de usar y comprender.
- **RNF-9 Compatibilidad:** El sistema debe ser compatible con otros sistemas.
- **RNF-10 Cumplimiento de la normativa:** El sistema debe cumplir con todas las leyes y reglamentos aplicables.

B.4. Especificación de requisitos

Diagrama de casos de uso

Actores

Solo se considerarán como actores al sistema y al usuario que interacciona con este.

Casos de uso

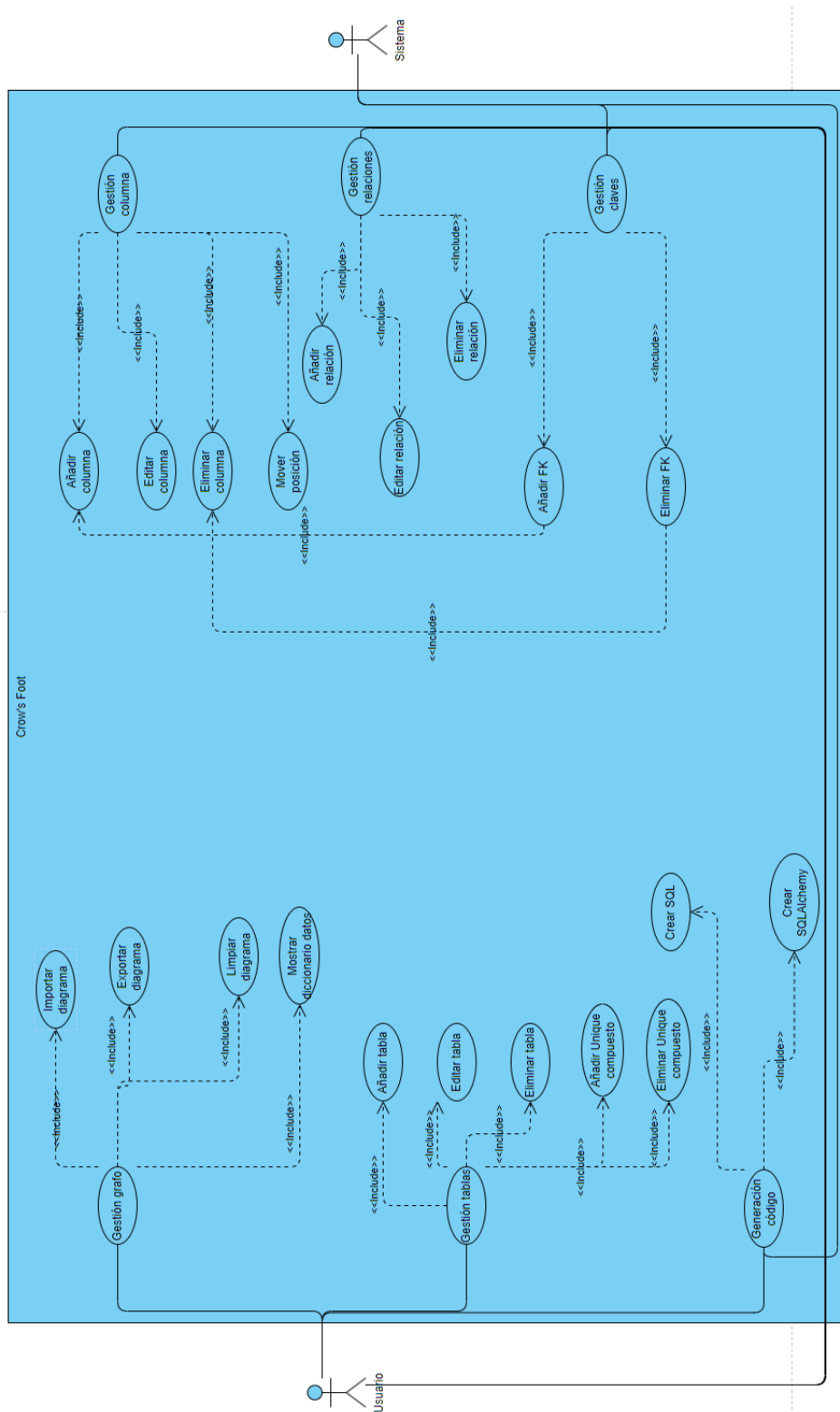


Figura B.1: Diagrama de Casos de Uso

CU-01	Gestión del grafo
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-1, RF-1.1, RF-1.2, RF-1.3
Descripción	Permite al usuario gestionar los elementos que se muestran en su diagrama
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none">1. El usuario accede a la aplicación2. El sistema muestra al usuario las opciones que dispone
Postcondición	El usuario puede diseñar su diagrama sin problemas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.1: CU-01 Gestión del grafo

CU-02	Importar diagrama
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-1, RF-1.1, RF-1.2
Descripción	Permite al usuario importar un diagrama creado con anterioridad y exportado a un fichero XML
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible ■ El fichero XML está formateado correctamente
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario pulsa el botón <i>Importar XML</i>. 3. El sistema muestra una ventana en la que importar el fichero XML. 4. El usuario selecciona un fichero XML almacenado en su PC. <ol style="list-style-type: none"> a) El usuario decide importar el diagrama, mostrando los elementos en la aplicación. b) El usuario decide no importar el diagrama.
Postcondición	El usuario obtiene un diagrama a partir del fichero XML importado
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la aplicación ■ Error al importar el diagrama desde el fichero
Importancia	Alta

Tabla B.2: CU-02 Importar diagrama

CU-03	Exportar diagrama
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-1, RF-1.1, RF-1.2
Descripción	Permite al usuario exportar el diagrama creado en el grafo a un fichero XML
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario añade elementos al grafo, creando su diagrama. 3. El usuario pulsa el botón <i>Exportar XML</i>. 4. El sistema genera el código XML correspondiente a los elementos del diagrama, mostrándolo en una ventana. <ol style="list-style-type: none"> a) Si el usuario pulsa el botón descargar, se mostrará una ventana donde introducir el nombre del fichero y este se descargará. b) El usuario puede cerrar la ventana sin descargar el fichero XML.
Postcondición	El usuario obtiene un fichero XML con los elementos del diagrama
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.3: CU-03 Exportar diagrama

CU-04	Limpiar diagrama
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-1, RF-1.4
Descripción	Permite al usuario eliminar todos los elementos añadidos a su diagrama
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none">1. El usuario accede a la aplicación.2. El usuario añade elementos al grafo, creando su diagrama.3. El usuario pulsa el botón <i>Borrar todo</i>.4. El sistema elimina todos los elementos que el usuario hubiera añadido al diagrama.
Postcondición	El usuario obtiene el código SQL correspondiente a su diagrama
Excepciones	Error al cargar la aplicación
Importancia	Baja

Tabla B.4: CU-04 Limpiar diagrama

CU-05	Mostrar diccionario de datos
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-1, RF-1.4
Descripción	Permite al usuario consultar un diccionario de datos con toda la información que haya añadido sobre sus las columnas de las tabas que se encuentren en el diagrama.
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario pulsa el botón de la barra inferior <i>Diccionario de datos</i>. 3. El sistema lista la información almacenada de las columnas de todas las tablas del diagrama.
Postcondición	El sistema muestra una ventana modal con el diccionario de datos
Excepciones	Error al cargar la aplicación
Importancia	Baja

Tabla B.5: CU-05 Mostrar diccionario de datos

CU-06	Gestión de tablas
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.1, RF-2.2, RF-2.3, RF-2.4, RF-2.5
Descripción	Permite al usuario gestionar las tablas que forman parte de su diagrama
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none">1. El usuario accede a la aplicación.2. El sistema muestra las opciones para trabajar con las tablas.
Postcondición	El usuario es capaz de manejar las tablas de su diagrama
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.6: CU-06 Gestión de tablas

CU-07	Añadir tabla
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.1
Descripción	Permite al usuario añadir tablas a su diagrama
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario utiliza el icono en el apartado de elementos, arrastrando este sobre el diagrama. 3. El usuario introduce un nombre para la nueva tabla. 4. El sistema crea la tabla con una clave primaria y la añade al diagrama.
Postcondición	El usuario es capaz de añadir tablas al diagrama
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la aplicación. ■ El usuario ha soltado el icono tabla en un lugar que no corresponde con el diagrama.
Importancia	Alta

Tabla B.7: CU-07 Añadir tabla

CU-08	Editar tabla
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.2
Descripción	Permite al usuario editar las tablas de su diagrama
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una tabla de su diagrama. 3. El sistema muestra un panel lateral con las propiedades que puede modificar de elemento tabla. 4. El usuario modifica los parámetros que considere. 5. El sistema aplica los cambios.
Postcondición	El usuario es capaz de editar tablas del diagrama
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.8: CU-08 Editar tabla

CU-09	Eliminar tabla
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.3
Descripción	Permite al usuario eliminar las tablas de su diagrama
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una tabla de su diagrama. <ol style="list-style-type: none"> a) El usuario pulsa el botón <i>Borrar</i> de la barra superior. b) El usuario hace click derecho en la tabla <ul style="list-style-type: none"> ■ El sistema muestra un menú popup ■ El usuario pulsa el botón <i>Borrar</i>. 3. El sistema elimina del diagrama la tabla seleccionada.
Postcondición	El usuario es capaz de Eliminar tablas del diagrama
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.9: CU-09 Eliminar tabla

CU-010	Añadir Unique compuesto
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.4, RF-2.5
Descripción	Permite al añadir Uniques compuestos a sus tablas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama. ■ La tabla contiene, al menos, dos columnas.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una tabla de su diagrama. 3. El usuario accede al apartado <i>Datos</i> del panel lateral. 4. El usuario pulsa el botón añadir en el apartado <i>Unique compuesto</i>. 5. El sistema lista las columnas de la tabla. <ol style="list-style-type: none"> a) El usuario selecciona al menos dos columnas <ul style="list-style-type: none"> ■ El usuario pulsa el botón aceptar. ■ El sistema almacena el nuevo unique compuesto. b) El usuario selecciona únicamente una columna <ul style="list-style-type: none"> ■ El sistema muestra una alerta advirtiéndole al usuario que debe seleccionar dos columnas. c) El usuario pulsa el botón cancelar o cierra la ventana, cancelando la operación
Postcondición	El sistema almacena un unique compuesto
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.10: CU-10 Añadir Unique compuesto

CU-11	Eliminar Unique compuesto
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-2, RF-2.4, RF-2.5
Descripción	Permite al usuario eliminar los uniques compuestos que haya creado anteriormente
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama ■ El usuario ha añadido, al menos, un unique compuesto a la tabla.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una tabla de su diagrama. 3. El usuario accede al apartado <i>Datos</i> del panel lateral. 4. El sistema lista los uniques compuestos que se han creado para dicha tabla. 5. El usuario pulsa el botón eliminar asociado a uno de los posibles unique compuestos. 6. El sistema elimina el unique compuesto correspondiente.
Postcondición	El usuario es capaz de eliminar los Unique compuestos que haya almacenado anteriormente
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.11: CU-11 Eliminar Unique compuesto

CU-12	Gestión de columnas
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-3, RF-3.1, RF-3.2, RF-3.3, RF-3.4
Descripción	Permite al usuario gestionar las columnas que forman parte de las tablas de su diagrama
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none">1. El usuario accede a la aplicación.2. El sistema muestra al usuario las opciones para trabajar con columnas
Postcondición	El usuario es capaz de gestionar las columnas de las tablas de sus diagramas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.12: CU-12 Gestión de columnas

CU-13	Añadir columna
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-3, RF-3.1
Descripción	Permite al usuario añadir columnas a las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. <ol style="list-style-type: none"> a) El usuario arrastra el icono columna sobre una tabla. <ul style="list-style-type: none"> ■ El usuario introduce el nombre de la columna nueva. b) El usuario hace click derecho en una tabla. <ul style="list-style-type: none"> ■ El sistema muestra un menú popup ■ El usuario pulsa el botón añadir columna 2. El sistema añade una columna a la tabla seleccionada
Postcondición	El usuario es capaz de añadir columnas a las tablas de sus diagramas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.13: CU-13 Añadir columna

CU-14	Editar columna
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-3, RF-3.2
Descripción	Permite al usuario editar columnas a las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una columna de una tabla en el diagrama. 3. El sistema muestra un panel lateral con las propiedades que puede modificar del elemento columna. 4. El usuario modifica los parámetros que considere. 5. El sistema aplica los cambios.
Postcondición	El usuario es capaz de editar columnas de sus diagramas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.14: CU-14 Editar columna

CU-15	Eliminar columna
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-3, RF-3.3
Descripción	Permite al usuario eliminar columnas a las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una columna de una tabla en el diagrama. <ol style="list-style-type: none"> a) El usuario pulsa el botón <i>Borrar</i> de la barra superior. b) El usuario hace click derecho en la columna. <ul style="list-style-type: none"> ■ El sistema muestra un menú popup. ■ El usuario pulsa el botón <i>Borrar</i>. 3. El sistema elimina del diagrama la columna seleccionada.
Postcondición	El usuario es capaz de eliminar columnas de sus diagramas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.15: CU-15 Eliminar columna

CU-16	Mover posición
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-3, RF-3.4
Descripción	Permite al usuario añadir columnas a las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tabla al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario selecciona una columna de una tabla en el diagrama. 3. El usuario hace click derecho en la columna. 4. El sistema muestra un menú popup. <ol style="list-style-type: none"> a) El usuario pulsa el botón <i>Subir posición</i>. b) El usuario pulsa el botón <i>Bajar posición</i>. 5. El sistema comprueba que puede mover la columna <ol style="list-style-type: none"> a) El sistema coloca la columna en la nueva posición. b) El sistema no mueve la columna.
Postcondición	El usuario es capaz de mover columnas de posición dentro de las tablas.
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.16: CU-16 Mover posición

CU-17	Gestión de relaciones
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-4, RF-4.1, RF-4.2, RF-4.3
Descripción	Permite al usuario gestionar las relaciones entre las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El sistema muestra al usuario las opciones para trabajar con relaciones
Postcondición	El usuario es capaz de gestionar las relaciones entre las tablas de los diagramas.
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.17: CU-17 Gestión de relaciones

CU-18	Añadir relación
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-4, RF-4.1
Descripción	Permite al usuario añadir relaciones entre las tablas de sus diagramas
Precondición	<ul style="list-style-type: none">■ La aplicación se encuentra disponible.■ El usuario ha añadido, al menos, dos tablas al diagrama
Acciones	<ol style="list-style-type: none">1. El usuario accede a la aplicación.2. El usuario mantiene pulsado sobre una tabla.3. El usuario arrastra sin soltar sobre otra tabla.4. El sistema crea una relación entre las dos tablas.
Postcondición	El usuario es capaz de añadir relaciones entre las tablas de los diagramas.
Excepciones	<ul style="list-style-type: none">■ Error al cargar la aplicación■ El usuario suelta el ratón en algún elemento que no es una tabla
Importancia	Alta

Tabla B.18: CU-18 Añadir relación

CU-19	Editar relación
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-4, RF-4.2
Descripción	Permite al usuario editar las relaciones entre las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama ■ El usuario ha creado, al menos, una relación entre dos tablas.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario hace click en una relación. 3. El sistema muestra un panel lateral con las propiedades que puede modificar del elemento relación. 4. El usuario modifica los parámetros que considere. 5. El sistema aplica los cambios.
Postcondición	El usuario es capaz de editar los datos de relaciones
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.19: CU-19 Editar relación

CU-20	Eliminar relación
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-4, RF-4.3
Descripción	Permite al usuario eliminar relaciones entre las tablas de sus diagramas
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama ■ El usuario ha creado, al menos, una relación entre dos tablas.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario hace click en una relación. <ol style="list-style-type: none"> a) El usuario pulsa el botón <i>Borrar</i> de la barra superior. b) El usuario hace click derecho en la columna. <ul style="list-style-type: none"> ■ El sistema muestra un menú popup. ■ El usuario pulsa el botón <i>Borrar</i>. 3. El sistema elimina del diagrama la relación seleccionada. 4. El sistema elimina las columnas correspondientes a claves foráneas asociadas con la relación.
Postcondición	El usuario es capaz de eliminar las relaciones entre dos tablas
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.20: CU-20 Eliminar relación

CU-21	Gestión de claves
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-5, RF-5.1, RF-5.2
Descripción	Permite al sistema gestionar las claves foráneas asociadas a cada tabla.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El sistema gestiona las claves foráneas cuando sea necesario.
Postcondición	El sistema es capaz de gestionar las claves foráneas asociadas a cada tabla.
Excepciones	Error al cargar la aplicación
Alta	Importancia

Tabla B.21: CU-21 Gestión de claves

CU-22	Añadir clave foránea
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-5, RF-5.1, RF-3.1
Descripción	Permite al sistema añadir las claves foráneas en las tablas que sea necesario.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario crea una relación entre dos tablas. 3. El sistema añade las claves foráneas en las tablas que se necesario, con las propiedades de la clave primaria.
Postcondición	El sistema es capaz de añadir claves foráneas como columnas.
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la aplicación ■ La tabla de la que se pretende obtener la clave primaria no cuenta con una
Importancia	Alta

Tabla B.22: CU-22 Añadir clave foránea

CU-23	Eliminar clave foránea
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-5, RF-5.1, RF-3.3
Descripción	Permite al sistema eliminar las claves foráneas en las tablas que sea necesario.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, dos tablas al diagrama. ■ El usuario ha añadido, al menos, una relación entre dos tablas.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario elimina una columna que representa una clave primaria. 3. El sistema elimina automáticamente las claves foráneas asociadas con dicha clave primaria.
Postcondición	El sistema es capaz de eliminar claves foráneas automáticamente.
Excepciones	Error al cargar la aplicación
Importancia	Alta

Tabla B.23: CU-23 Eliminar clave foránea

CU-24	Generación de código
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-6, RF-6.1, RF-6.2
Descripción	Permite al sistema generar el código correspondiente al diagrama que ha creado el usuario para diferentes lenguajes.
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible. ■ El usuario ha añadido, al menos, una tablas al diagrama.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario utiliza uno de los botones para generar código. 3. El sistema genera el código correspondiente.
Postcondición	El sistema es capaz de generar código a partir del diagrama del usuario.
Excepciones	Error al cargar la aplicación
Alta	Importancia

Tabla B.24: CU-24 Generación de código

CU-25	Crear SQL
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-5, RF-5.1
Descripción	Permite al usuario generar código SQL a partir del diagrama creado en la aplicación
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible ■ Se han añadido elementos al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario añade elementos al grafo, creando su diagrama. 3. El usuario pulsa el botón <i>Mostrar SQL</i>. 4. El sistema genera el código SQL correspondiente a los elementos que se muestran en el diagrama.
Postcondición	El usuario obtiene el código SQL correspondiente a su diagrama
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la aplicación ■ No hay elementos en el diagrama
Importancia	Alta

Tabla B.25: CU-25 Crear SQL

CU-25	Crear SQLAlchemy
Versión	1.0
Autor	Juan Romera Pérez
Requisitos asociados	RF-5, RF-5.1
Descripción	Permite al usuario generar código para SQLAlchemy a partir del diagrama creado en la aplicación
Precondición	<ul style="list-style-type: none"> ■ La aplicación se encuentra disponible ■ Se han añadido elementos al diagrama
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario añade elementos al grafo, creando su diagrama. 3. El usuario pulsa el botón <i>Mostrar SQLAlchemy</i>. 4. El sistema genera el código SQLAlchemy correspondiente a los elementos que se muestran en el diagrama.
Postcondición	El usuario obtiene el código SQLAlchemy correspondiente a su diagrama
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la aplicación ■ No hay elementos en el diagrama
Importancia	Alta

Tabla B.26: CU-25 Crear SQLAlchemy

Apéndice C

Especificación de diseño

C.1. Introducción

En esta especificación de diseño se detallan las decisiones de diseño tomadas durante el proceso de desarrollo de la aplicación. A continuación, se analizan los datos que va a manejar la aplicación, sus detalles procedimentales y la arquitectura utilizada.

C.2. Diseño de datos

En este apartado describiremos las estructuras de datos utilizadas en la aplicación para representar los diagramas.

Entidades

En la aplicación se maneja la representación de los diagramas utilizando tres entidades, que describiremos a continuación.

Las entidades aquí descritas corresponden a los objetos almacenados en los objetos celda de la librería *mxGraph*. En la descripción se incluirán diferentes atributos de los objetos *mxCell*, ya que se han utilizado durante el desarrollo y no se almacenan, al no ser necesario. Dichos atributos se diferenciarán correctamente (escribiéndolos en cursiva) de los de nuestros objetos. No se incluirán todos los atributos de esta clase, sólo los relevantes para nuestra aplicación.

Tabla

La entidad *Tabla* representa una tabla en el diagrama. A cada tabla se le asigna un nombre y puede contener diferentes columnas. Los atributos utilizados son:

- **Id:** Id asociado a la celda que representa la tabla.
- **name:** Nombre de la tabla.
- **uniqueComp:** Array compuesto por arrays de enteros, que almacenan los identificadores de las columnas asociadas a cada unique compuesto.
- **gradient:** Booleano que indica si mostrar degradado o no en la tabla.
- **geometry:** Geometría asociada a la celda de la tabla.
- **style:** Estilo asociado a la celda de la tabla.

Columna

La entidad *Columna* representa una columna dentro de una tabla del diagrama. Cada columna tiene asociado un nombre, su tipo y diferentes propiedades correspondientes con la base de datos. Además, tiene asociada una referencia a la tabla que la contiene (su celda padre en mxGraph) y, en caso de que la columna sea una clave foránea, tendrá asociada una referencia a la clave primaria y a la relación correspondientes. Los atributos de este objeto son:

- **Id:** Id asociado a la celda que representa la columna.
- **parent:** Referencia a la celda padre que contiene la columna, en este caso, una tabla del diagrama.
- **name:** Nombre de la columna.
- **type:** Tipo de la columna.
- **defaultValue:** Valor que debe tomar por defecto la columna.
- **primaryKey:** Indica si la columna es clave primaria.
- **foreignKey:** Indica si la columna es clave foránea.
- **autoIncrement:** Indica si el valor de la columna se debe auto incrementar.

- **notNull:** Indica si el valor de la columna no puede ser nulo.
- **unique:** Indica si el valor de la columna no se puede repetir.
- **desc:** Almacena la descripción asociada a la columna.
- **titulo:** Almacena el título de la columna.
- **relacionAsociada:** Almacena el id de la relación que tiene asociada, en caso de que la columna sea clave foránea.
- **pkAsociada:** Almacena el id de la columna de otra tabla que tiene asociada, en caso de que la columna sea clave foránea.
- **geometry:** Geometría asociada a la celda de la columna.
- **style:** Estilo asociado a la celda de la columna.

Relación

La entidad *Relación* representa la conexión entre dos tablas que se relacionan entre sí. Las relaciones tendrán dos tablas asociadas, con un símbolo para cada extremo. Los atributos del objeto asociado son:

- **Id:** Id asociado a la celda que representa la relación.
- **source:** Referencia a la tabla que se encuentra en el lado origen de la relación. En esta tabla se encontrarán las claves foráneas asociadas a la relación.
- **target:** Referencia a la tabla que se encuentra en el lado destino de la relación.
- **name:** Nombre de la relación.
- **clavesForaneas:** Array de números compuesto por los ids de las columnas que corresponden a las claves foráneas asociadas a la relación.
- **startArrow:** Indica el símbolo de la notación de patas de cuervo que se debe mostrar en el lado origen de la relación.
- **endArrow:** Indica el símbolo de la notación de patas de cuervo que se debe mostrar en el lado destino de la relación.
- **geometry:** Geometría asociada a la celda de la relación.
- **style:** Estilo asociado a la celda de la relación.

Diagrama de clases

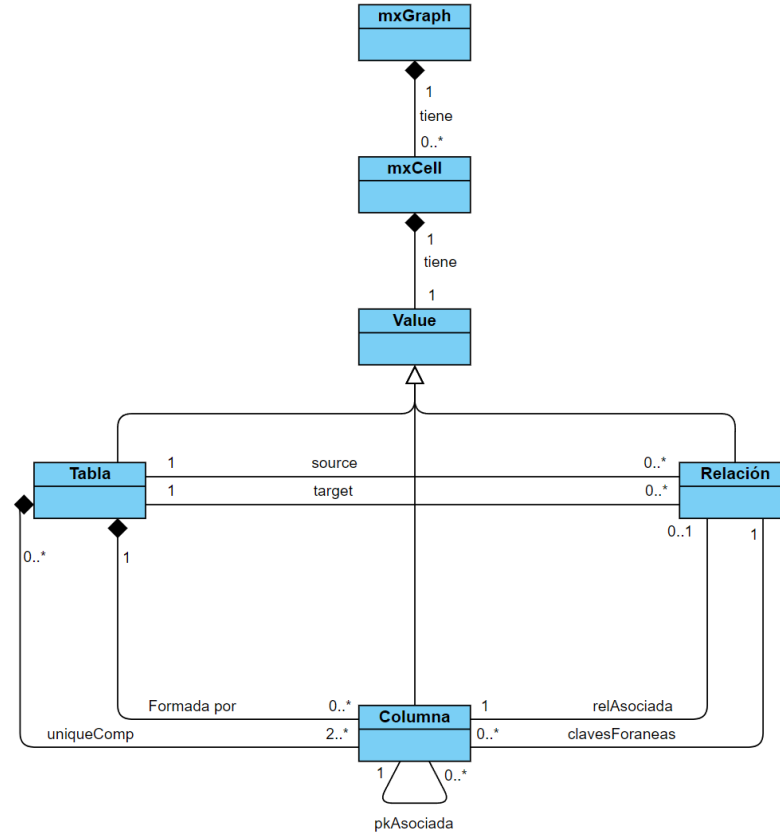


Figura C.1: Diagrama de clases

En el diagrama de clases se han incluido las clases *Tabla*, *Relación* y *Columna* como especializaciones de *value*, el atributo en el que se almacenan, ya que, aunque no tengan atributos o comportamientos similares, cumplen la misma función dentro de los objetos celda (*mxCell*). Se ha decidido incluir también que un elemento *mxGraph* está compuesto por 0 o muchos elementos *mxCell*, siendo estas dos las clases más relevantes para el tema tratado.

C.3. Diseño procedimental

Introducción

En este apartado se describen los procedimientos y métodos implementados en la aplicación para manejar los datos y la lógica de negocio. Se

detallan las funciones principales, su propósito y cómo interactúan entre sí, junto a los diagramas necesarios para visualizar dichos procesos.

Configuración del grafo

La configuración del grafo se realiza desde la función *main*. Desde esta función, configuraremos los parámetros básicos del grafo. Añadimos los símbolos personalizados, correspondientes a la notación de patas de cuervo, configuramos las propiedades y estilos de los elementos del diagrama e inicializamos los componentes de la interfaz de usuario.

Añadir Tabla/Columna

Para añadir una tabla o una columna al diagrama utilizaremos la función asociada al icono correspondiente del panel lateral, añadido desde la función *addSidebarIcon*. En esta función añadiremos los iconos de tabla y columna, transformándolos en elementos *draggable* y añadiremos una función con la que manejaremos la incorporación de nuevos elementos al diagrama, después de haber comprobado que el drop del elemento se puede completar.

Una vez comprobado que el destino para el nuevo elemento es válido, ejecutaremos nuestra función. El sistema comprobará donde ha ocurrido el evento de soltado del ratón. En función del prototipo indicado, el nuevo elemento será tabla o columna.

1. Si es *columna*, comprobaremos si debajo del punto obtenido hay una tabla. Si hay una tabla, se continua con el proceso, ajustando el punto para que se ajuste a la geometría de la tabla padre. Solicitaremos un nombre para la columna al usuario y añadiremos la columna a la tabla.
2. Si es una *tabla*, solicitaremos un nombre al usuario y añadiremos una nueva tabla en el punto obtenido.

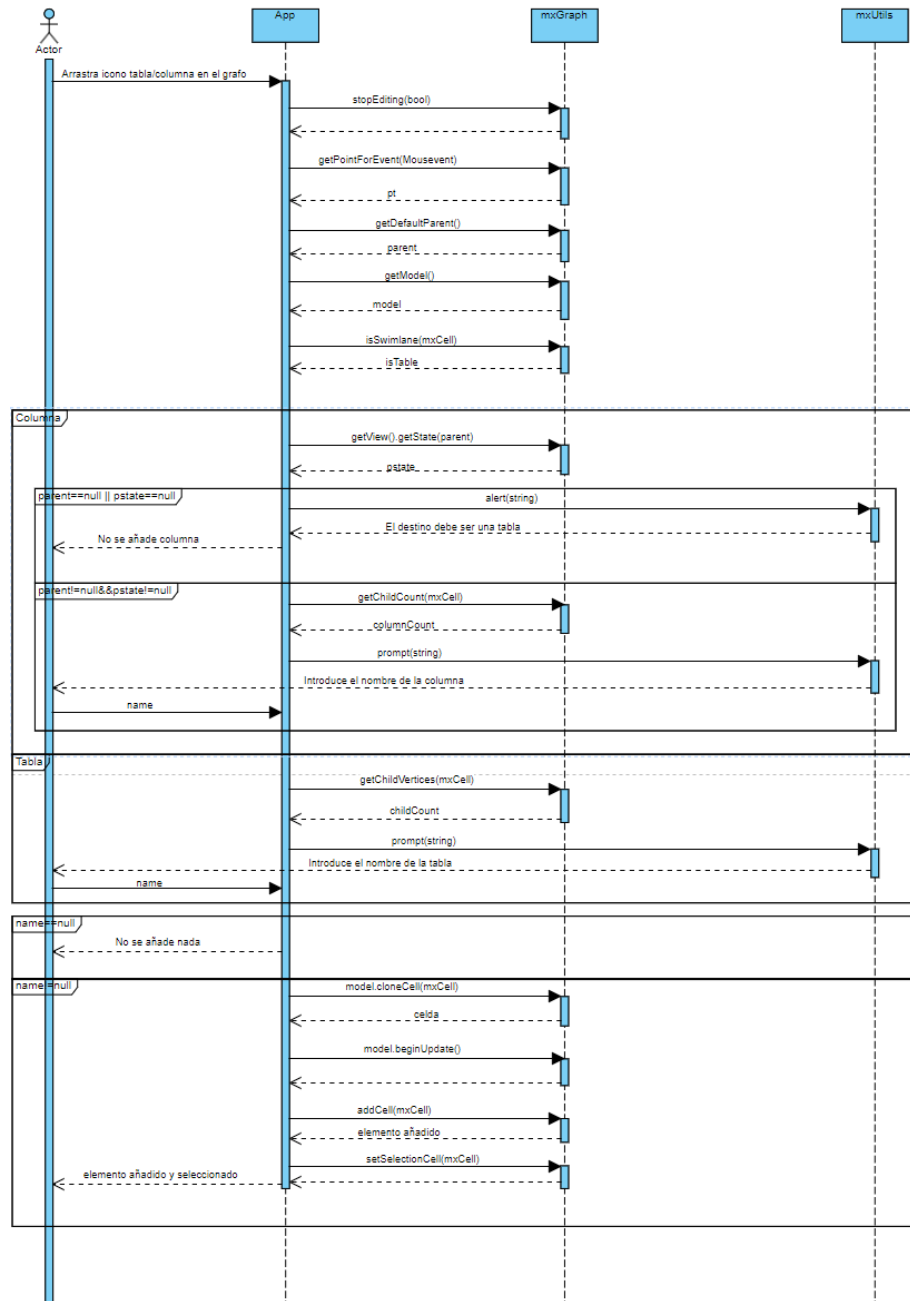


Figura C.2: Diagrama secuencia añadir Tabla/Columna

Añadir relación

Utilizamos la función *addEdge* de *mxGraph* para crear una nueva relación entre dos tablas. Primero, comprobamos que la tabla destino del enlace tiene

clave primaria, pues está se añadirá como clave foránea en la tabla origen. Si tiene clave primaria, creamos un nuevo objeto *Relacion* y llamamos a la función *insertarNuevaRelacion*. En esta función, configuramos la nueva relación con la función *editarRelacion* añadiéndola al diagrama, y añadimos las claves foráneas necesarios con la función *addClaveForanea*.

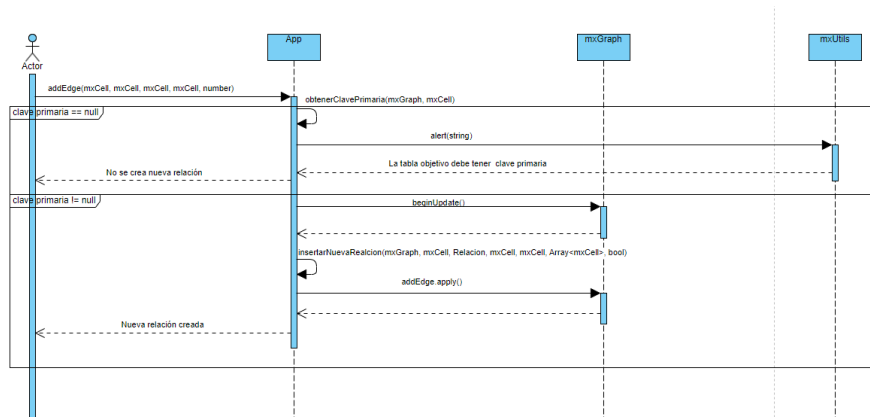


Figura C.3: Diagrama secuencia añadir relación

Editar elemento

Cada vez que cambia el elemento seleccionado en el diagrama, el listener incluido en *getSelectionModel* de *mxGraph* cambiará los elementos del panel lateral de propiedades. Este panel está compuesto por el apartado de *Estilos* y el apartado de *Datos*. En el caso de que el elemento seleccionada sea una celda de *mxGraph*, es decir, una tabla, columna o relación, el panel se mostrará. Si se deja de seleccionar un elemento de este tipo el panel se oculta.

Estilos

Cada vez que cambie la selección se llama a la función *configurarTabEstilos*, desde la que se establecerán los valores para cada elemento obteniendo dichos valores desde el atributo *style* de la celda seleccionada, mostrando u ocultando elementos si fuera necesario con la función *setEstilosIniciales*. A cada elemento de configuración de estilos se le añade un listener desde la función *setListenersEstilos*, que realizará las comprobaciones necesarias y modificará la apariencia de la celda. Una vez se cambia de selección, se eliminan todos estos listeners para no modificar varias celdas a la vez.

Datos

Para cada tipo de elemento se creará una tabla con los parámetros que se pueden modificar desde la función *showProperties*. La tabla tiene un listener que modifica la información de la celda cada vez que ocurre un cambio.

A continuación, se describe el proceso en los cambios más relevantes que pueden ocurrir.

Actualizar relaciones

Desde el panel de datos podemos editar los símbolos de la notación de patas de cuervo que aparecen en las relaciones. Con cada cambio se actualizarán las claves con la función *actualizarClaves*, ajustando la tabla en la que se muestran y sus parámetros. Desde la función actualizar claves se eliminará la celda correspondiente al enlace actual y se añadirá una nueva relación con los nuevos parámetros utilizando la función *insertarNuevaRelacion*. En caso de que sea necesario, se insertará una tabla intermedia con la función *obtenerTablaIntermedia*.

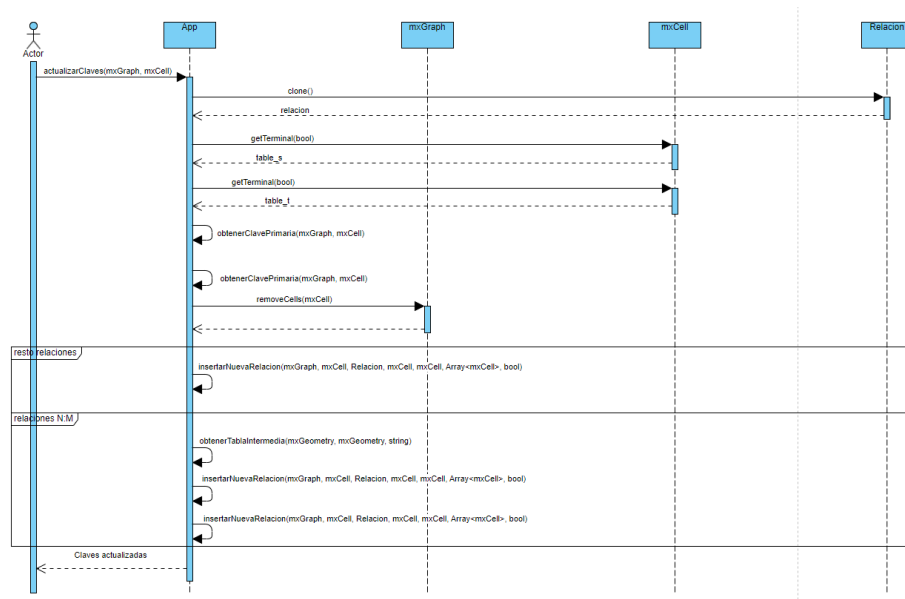


Figura C.4: Diagrama secuencia actualizar relaciones

Cambio clave primaria

Cuando una columna cambie su atributo de clave primaria, se actualizarán las claves de las relaciones que pudiera tener asociada su tabla padre utilizando la función *handleCambioClavePrimaria*. Desde esta función se

obtendrán todas las relaciones entrantes a la tabla padre, actualizando dichas relaciones con la nueva PK utilizando la función *actualizarClaves*.

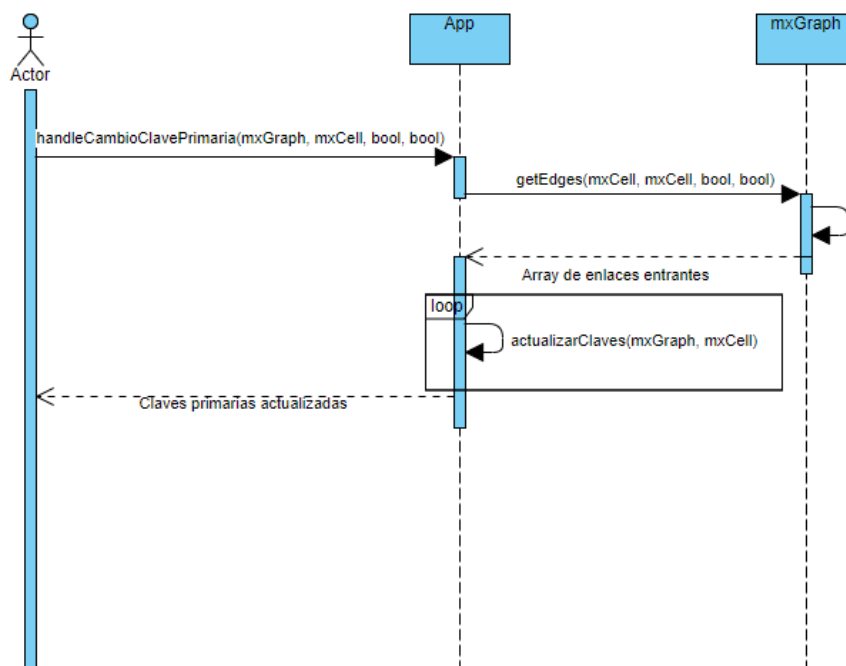


Figura C.5: Diagrama secuencia cambio clave primaria

Añadir Unique compuesto

Cuando se quiera añadir un nuevo Unique compuesto, utilizando la función *addUniqueComp*, el sistema comprobará que la tabla tiene al menos dos columnas. Después se mostrará un formulario con las columnas disponibles a añadir. Al finalizar, se almacena un array con los ids de las columnas seleccionadas en la tabla correspondiente.

Mover posición columna

Cada vez que se quiera mover de posición una columna, se comprobará que hay una celda disponible en la dirección en la que se pretende mover. En caso de que existe, se llamará a la función *moverPosicionColumna*. Se actualizarán las posibles referencias que pudieran tener las columnas con las funciones *intercambioIds*, *actualizarRefPK* y *intercambioIdsUnique*, y las celdas asociadas se intercambiarán los atributos *value* y *style*.

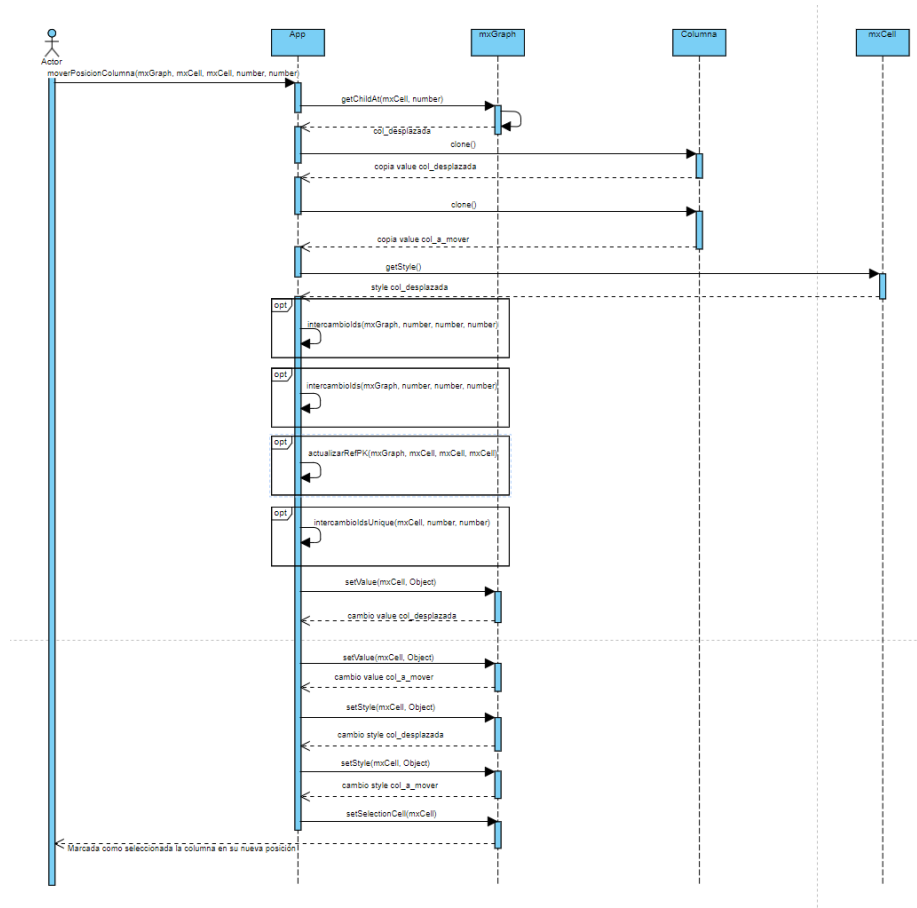


Figura C.6: Diagrama secuencia mover columna

Generación de código

El proceso seguido para los dos tipos de generación de código disponibles es muy similar, por lo que se describirá de forma general. Cuando se utilice uno de los botones de generación de código se utilizará la función *createSql* o *createSqlAlchemy*, que recorren todas las tablas del diagrama llamando a *addTablaSql* o *addTablaSqlAlchemy* para cada una. Desde estas funciones, se introduce el código correspondiente a la creación de la tabla, recorriendo todas sus columnas e incluyendo sus datos con *addColumnaSql* o *addColumnaSqlAlchemy*. Al final de cada columna se añaden claves primarias, en el caso de SQL, claves foráneas y uniques compuestos y se une todo el contenido obtenido al final.

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este anexo se presenta la documentación técnica de programación, que incluye una descripción de la estructura de directorios utilizada, una explicación para el futuro trabajo desarrollo de la aplicación

D.2. Estructura de directorios

La estructura de directorios utilizada que podemos ver en el repositorio es la siguiente:

- `/`: Contiene ficheros de configuración, además del fichero README.
- `/CrowsFoot`: Directorio principal de la aplicación.
- `/CrowsFoot/editors`: Imágenes y ficheros de configuración, heredados de la librería.
- `/CrowsFoot/editors/config`: Ficheros de configuración en formato XML para trabajar con la librería mxGraph.
- `/CrowsFoot/editors/images`: Imágenes utilizadas en diferentes botones dentro de la aplicación.
- `/CrowsFoot/images`: Imágenes variadas para la aplicación.

- **/CrowsFoot/mxgraph**: Directorio principal de la librería mxGraph.
- **/CrowsFoot/src**: Directorio donde encontramos los elementos principales del proyecto.

D.3. Manual del programador

En el siguiente apartado se pretende explicar a futuros programadores como trabajar con la aplicación. Necesitaremos instalar los siguientes programas.

IDE

El entorno de trabajo que utilizaremos es un IDE que permita *Javascript*, por lo que la mayoría de entornos de desarrollo nos pueden servir. En nuestro caso hemos utilizado **VSCode**.

Git

Necesitamos tener instalado *Git* para hacer uso del control de versiones y poder obtener el repositorio de *Github* de forma local realizando un clonado de este. *Git GUI* nos permite trabajar con Git tanto con una interfaz de usuario común, como con línea de comandos como se acostumbra a trabajar con Git.

Navegador

Puesto que se trata de una aplicación web, necesitaremos un navegador moderno (Chrome, Firefox, Opera, etc.) que nos permita ejecutar la aplicación.

D.4. Compilación, instalación y ejecución del proyecto

Compilación

Javascript es un lenguaje el cual se compila en tiempo real o JIT[7], por lo que no necesitamos ningún compilador, el navegador que utilicemos se encargará de realizar el compilado en tiempo real.

Instalación

Para instalar el proyecto clonaremos el repositorio desde *Github*. Haremos uso de la herramienta Git GUI, tanto de su versión de interfaz como de línea de comandos, para obtener el repositorio. Además, se recomienda tener instalada la extensión *Live Server*, o similares, la cual nos permite desplegar un servidor local donde se aloja nuestra aplicación. De esta forma podremos ver el comportamiento de la aplicación en un despliegue real, además de poder observar posibles errores que no sería posible de otra forma.

Con la interfaz gráfica simplemente completando los campos que nos pide la herramienta podremos obtener el repositorio.

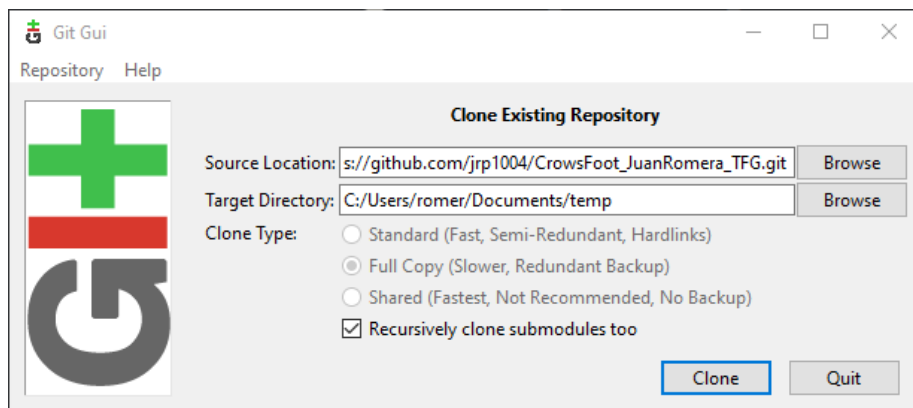


Figura D.1: Clonación con Git GUI

Una vez importado el proyecto a nuestro sistema, podemos utilizar también esta herramienta para realizar las operaciones con Git que necesitemos, aunque no es indispensable, es otra opción que tiene esta herramienta.

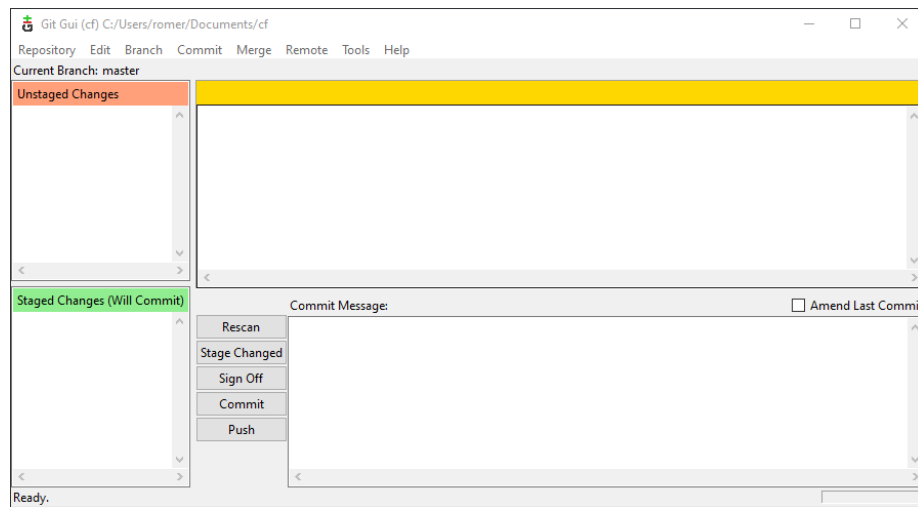


Figura D.2: Git GUI

Para clonar el repositorio mediante la línea de comandos utilizaremos el comando `git clone`. Desde la línea de comandos accedemos al directorio donde deseamos colocar la copia del repositorio, introducimos el comando y la herramienta realizará la clonación.

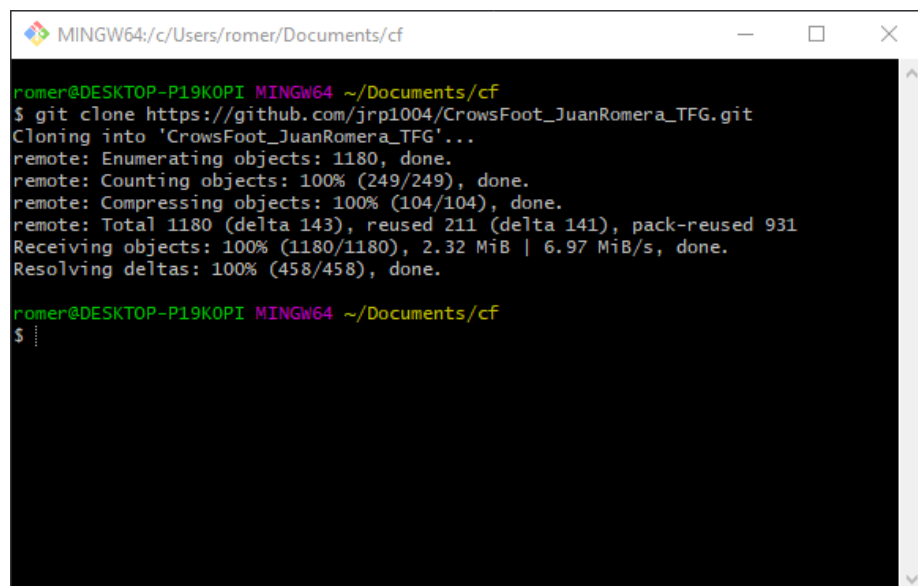


Figura D.3: Clonación Git Bash

Añadir características al proyecto

Una vez obtenida una copia del repositorio, en caso de que queramos realizar cambios en la aplicación, crearemos una nueva rama desde la rama *develop* con el nombre de la nueva funcionalidad a implementar y comenzaremos a trabajar desde ahí.

Una vez realizados los commits y pasando las pruebas de calidad necesarias, se incorporarán los cambios a la rama *develop* realizando un merge.

Ejecución

Para ejecutar el proyecto disponemos de dos formas.

Preview Vercel

Podemos utilizar las previews que nos proporciona *Vercel*, las cuales nos muestran el último estado de la aplicación a partir del último push realizado al repositorio. La contra de utilizar este método es que necesitamos realizar el push al repositorio para poder visualizar los cambios, lo cual no nos permite ver los cambios que realizamos sobre la aplicación en tiempo real.

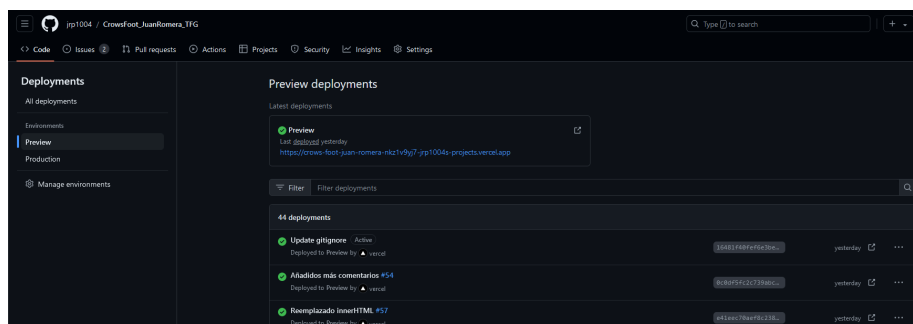


Figura D.4: Preview Vercel

Para acceder la preview, podemos hacerlo desde el repositorio en *Github*, en el apartado *Deployments* donde nos llevará a la ventana que vemos en la parte superior, con un enlace al despliegue de la preview.

Live Server

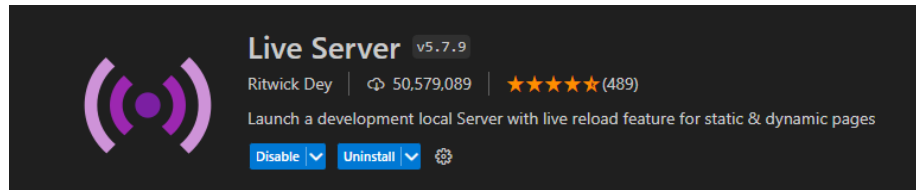


Figura D.5: Live Server

Live Server es una extensión del IDE VSCode que nos permite desplegar un servidor local con nuestra aplicación. Este servidor se recargará automáticamente cada vez que realicemos un cambio sobre el código, lo cual nos permite ver de forma instantánea el impacto de nuestros cambios sobre la aplicación.

Para desplegar el servidor, simplemente haremos click derecho sobre el fichero html y pulsaremos en la opción *Open with Live Server*.

Como contra, este requerirá de algo más de potencia de nuestra máquina, pues es la encargada de ejecutar la aplicación.

Apéndice E

Documentación de usuario

E.1. Introducción

En este manual se detalla como puede el usuario acceder y trabajar con la aplicación.

E.2. Requisitos de usuarios

Al tratarse de una aplicación web, el usuario necesita disponer de un PC con navegador instalado que le permita poder acceder a esta. Todo se ejecuta desde la web, el usuario no necesita disponer de nada más.

La aplicación está pensada para ser ejecutada desde un ordenador, aunque es posible, no se recomienda acceder desde un dispositivo móvil, ya que ni la distribución de la interfaz está pensada para una pantalla vertical, ni se dispone de soporte para controles táctiles.

E.3. Manual del usuario

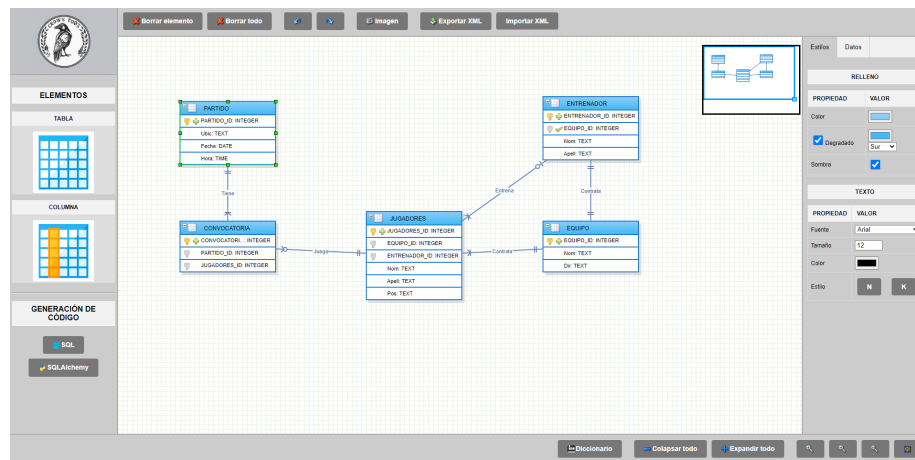


Figura E.1: Aplicación

Elementos del diagrama

Lo más importante a la hora de diseñar nuestros diagramas son los elementos que podemos utilizar, en nuestro caso, las tablas, sus columnas y las relaciones existentes. En la aplicación disponemos de los siguientes elementos que podremos añadir a nuestros diagramas.

Tablas

Para añadir una tabla al diagrama, arrastraremos el icono correspondiente sobre el diagrama.

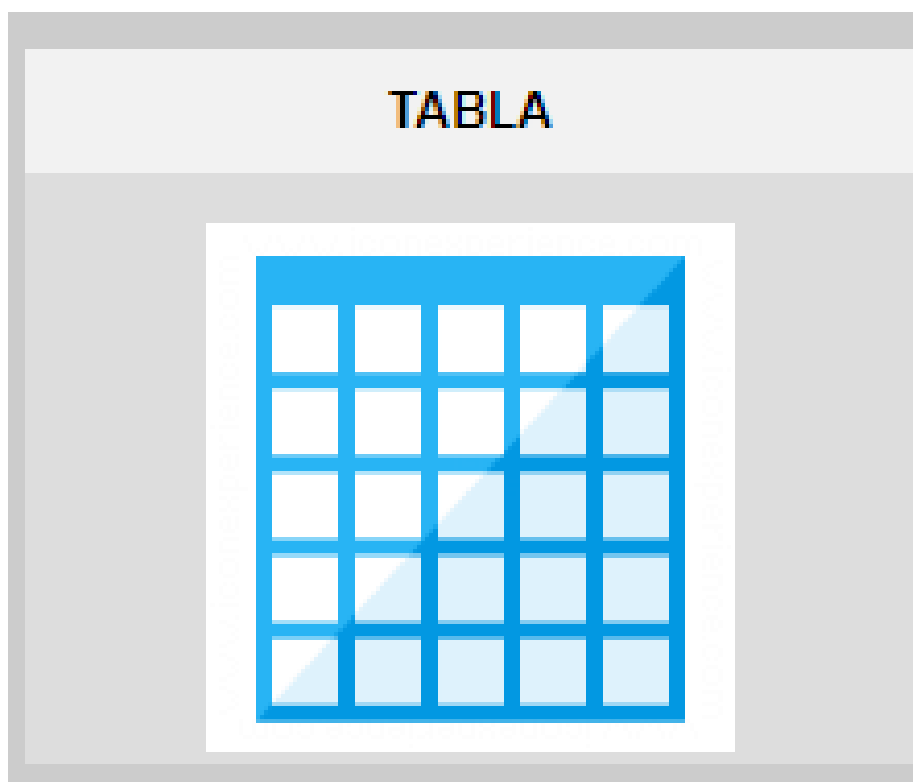


Figura E.2: Icono tabla

Una vez soltemos el icono sobre el grafo, la aplicación nos solicitará el nombre de la nueva tabla, proponiéndonos un nombre genérico.

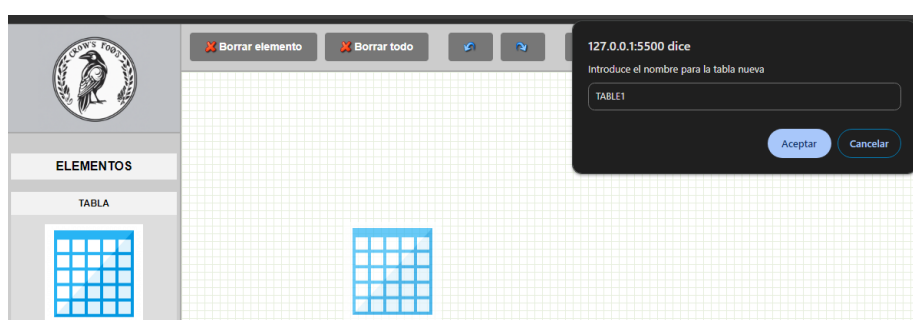


Figura E.3: Añadir tabla

Una vez demos nombre a la tabla esta se añade al diagrama.

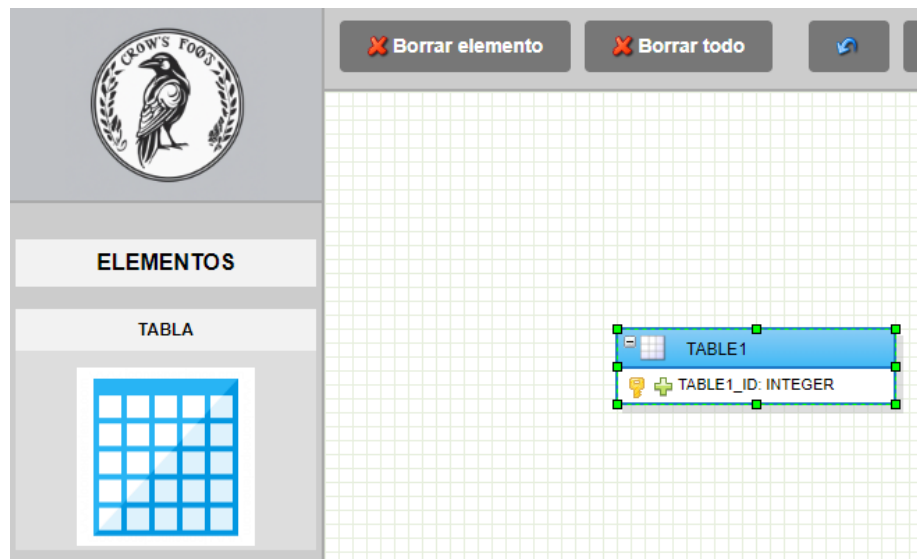


Figura E.4: Nueva tabla añadida

Como podemos ver, en el apartado superior se distingue el nombre de la tabla en el recuadro azul, representando a esta, además de una primera columna añadida de forma genérica a todas las tablas al crear, que corresponde con su posible clave primaria.

Columnas

Las columnas pueden ser añadidas a las tablas de dos formas:

Arrastrando el icono

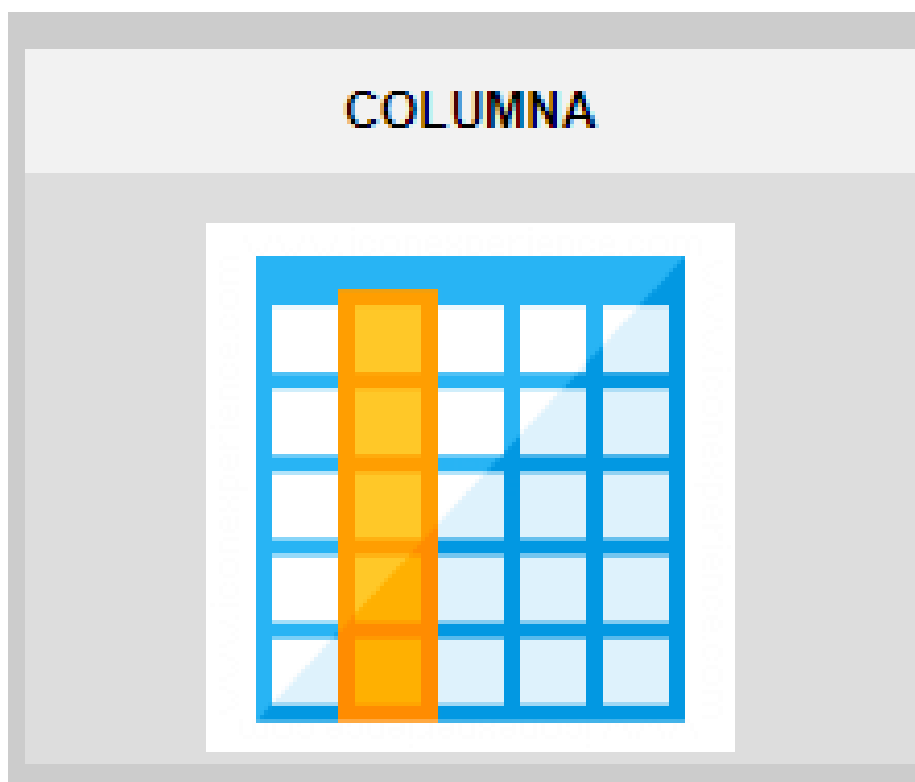


Figura E.5: Icono columna

Arrastrando el siguiente icono encima de una tabla crearemos una nueva columna para dicha tabla. Es importante señalar, ya que las columnas no pueden existir por si mismas, no podemos soltar el icono sobre el diagrama, tiene que ser siempre sobre una tabla.

Menú popup

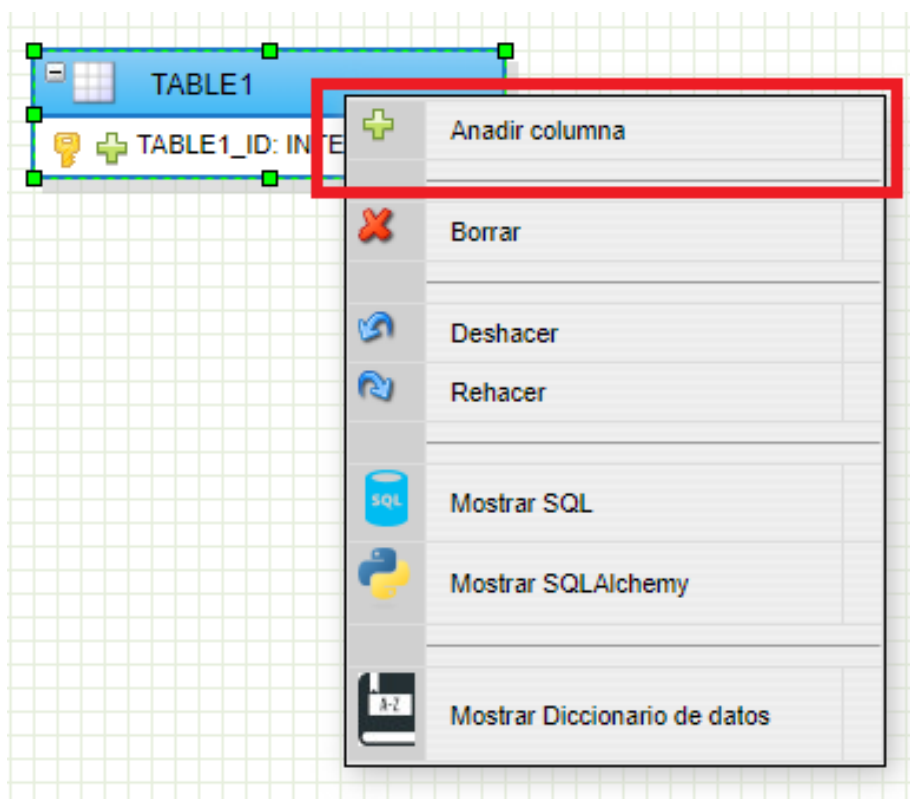


Figura E.6: Añadir columna desde menú popup

Haciendo click derecho en una tabla se desplegará un menú popup que nos presenta la opción "Añadir columna". Si pulsamos dicha opción se añadirá una columna a la tabla.

Atajo de teclado Cuando tengamos seleccionada una tabla, si pulsamos el atajo de teclado **F4**, podremos añadir una nueva columna a dicha tabla.

Una vez hemos completado una de estas acciones, de nuevo se nos solicitará un nombre para la columna, proporcionando uno genérico.

Relaciones

Para crear una relación entre dos tablas, colocaremos el cursor sobre una de las tablas, apareciendo el icono de una flecha cuando se pueda añadir la relación. Manteniendo pulsado el click izquierdo, arrastraremos el cursor hasta la tabla con la que queramos crear la relación. Cuando sea posible crear una relación con dicha tabla esta se resaltará de color verde.

Figura E.7: Icono flecha



Figura E.8: Soltar para añadir relación

Una vez soltamos el click izquierdo, se creará el nuevo enlace entre las dos tablas. Las relaciones nuevas siempre son 1:1, colocando la clave primaria de la tabla donde hemos soltado el cursor como clave foránea en la primera tabla. Además, se le añade un nombre genérico a la relación.

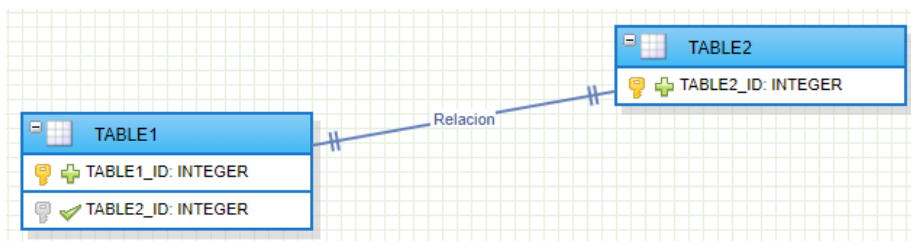
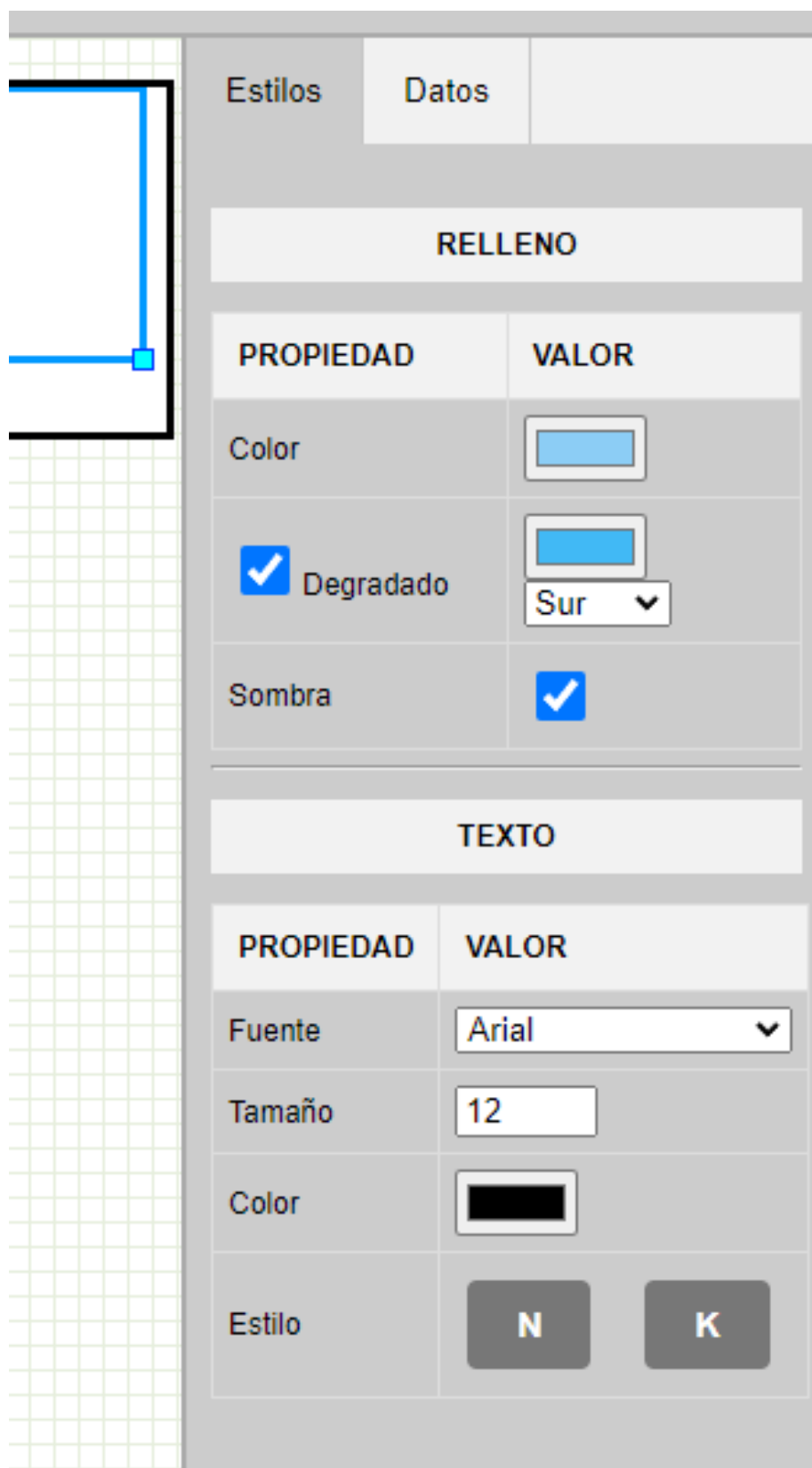


Figura E.9: Nueva relación creada

Panel propiedades

En el lado derecho de la ventana, disponemos de un panel que nos permite editar los diferentes elementos que aparecen en nuestro diagrama. Simplemente seleccionando uno de estos elementos, ya sea una tabla, columna o relación, el panel mostrará automáticamente las opciones que disponemos para modificar.



The image shows a software interface with a grid on the left and a properties panel on the right. The grid has a green background with a light gray grid pattern. A cell in the grid is selected, indicated by a blue border and a small blue square at its bottom-right corner. The properties panel is divided into two tabs: 'Estilos' (Styles) and 'Datos' (Data). The 'Estilos' tab is active. It contains two sections: 'RELLENO' (Fill) and 'TEXTO' (Text). The 'RELLENO' section has a table with two columns: 'PROPIEDAD' (Property) and 'VALOR' (Value). The 'TEXTO' section also has a table with two columns: 'PROPIEDAD' (Property) and 'VALOR' (Value).

PROPIEDAD	VALOR
Color	<input type="color" value="#add8e6"/>
<input checked="" type="checkbox"/> Degradado	<input type="color" value="#add8e6"/> Sur ▼
Sombra	<input checked="" type="checkbox"/>

PROPIEDAD	VALOR
Fuente	Arial ▼
Tamaño	12
Color	<input type="color" value="#000000"/>
Estilo	<input type="button" value="N"/> <input type="button" value="K"/>

Figura E.10: Panel propiedades

Este panel lateral se divide en dos pestañas con utilidades diferentes.

Pestaña Estilos

En este apartado podemos modificar la apariencia de los elementos del diagrama. Para ello, disponemos de dos secciones diferentes, una para el relleno y otra para el texto. Todas las opciones que se pueden modificar en este apartado únicamente tienen impacto estético.

Relleno

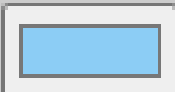
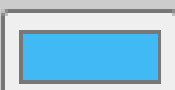
RELLENO	
PROPIEDAD	VALOR
Color	
<input checked="" type="checkbox"/> Degradado	 Sur ▼
Sombra	<input checked="" type="checkbox"/>

Figura E.11: Apartado relleno

En esta sección podemos modificar el color y el degradado, en el caso de las tablas y las columnas, y el color de la línea en el caso de las relaciones. Además, en el caso de las tablas, podemos escoger si mostramos su sombra.

Texto

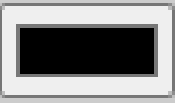

PROPIEDAD	VALOR
Fuente	Arial
Tamaño	12
Color	
Estilo	

Figura E.12: Apartado texto

Esta sección permite editar el estilo del texto que se muestra en cada elemento. Podemos modificar la fuente, pudiendo escoger entre las siguientes opciones.

- Arial
- Serif
- Sans-Serif
- Tahoma
- Verdana
- Dialog

- Times New Roman

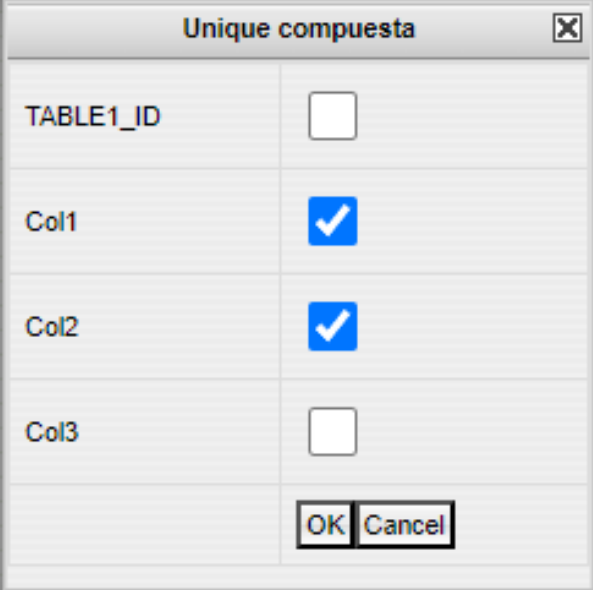
Además, es posible modificar el tamaño del texto, su color y seleccionar si mostrar el texto en negrita y en cursiva.

Pestaña Datos

Desde el apartado datos podemos modificar la información de cada uno de los elementos del diagrama.

Tabla

En las tablas del diagrama podemos modificar su nombre, además de gestionar los posibles Unique compuestos que pudiera tener esta tabla. Un Unique compuesto indica que la combinación de las columnas que lo forman es siempre única. Para añadir, un Unique compuesto necesitamos que la tabla seleccionada tenga al menos dos columnas. Una vez disponemos de estas dos columnas, pulsaremos el botón correspondiente, mostrando el siguiente panel.



El formulario 'Unique compuesta' muestra una lista de columnas de la tabla 'TABLE1_ID' con sus respectivos botones de selección. Las columnas 'Col1' y 'Col2' están seleccionadas, mientras que 'TABLE1_ID' y 'Col3' no lo están. Los botones 'OK' y 'Cancel' están ubicados en la parte inferior derecha del formulario.

Columna	Seleccionada
TABLE1_ID	<input type="checkbox"/>
Col1	<input checked="" type="checkbox"/>
Col2	<input checked="" type="checkbox"/>
Col3	<input type="checkbox"/>

OK Cancel

Figura E.13: Formulario para añadir unique compuesto

El panel listará todas las columnas que forman la tabla, pudiendo seleccionar todas las deseadas con un mínimo de dos. Una vez añadido el Unique compuesto, el sistema mostrará en la tabla todos aquellos que se hayan creado, además de un botón que nos permite eliminar cada uno de ellos.

Relación

Para las relaciones, en el apartado Datos podemos modificar también su nombre, además de la cardinalidad de la relación en cada uno de los extremos del enlace.

PROPIEDAD	VALOR
Nombre	Relacion
TABLE1	<div> <div>solo uno</div> <div>cero o uno</div> <div>cero o mas</div> <div>uno o mas</div> </div>
TABLE2	<div> <div>solo uno</div> <div>cero o uno</div> <div>cero o mas</div> <div>uno o mas</div> </div>

Figura E.14: Formulario editar relación

Cuando seleccionemos una cardinalidad nueva, el sistema automáticamente modificará el símbolo en el extremo correspondiente para mostrar el indicado. Además, se actualizarán de forma automática las claves foráneas, colocándolas en la tabla correspondiente junto con los parámetros necesarios. En el caso de las relaciones M:M, se creará una tabla intermedia representando esta relación, con una relación 1:M con las tablas correspondientes. Esta tabla se creará con una clave primaria propia incluyendo las claves primarias de las otras dos tablas como claves foráneas.

Columnas En las columnas, además del nombre como en el resto de elementos, podremos modificar los siguientes apartados.

- **Tipo:** Tipo de dato correspondiente a la columna. El tipo de dato es una cadena introducida por el usuario. Es responsabilidad de este que el tipo esté introducido correctamente.
- **Clave primaria:** Indica si la columna seleccionada es clave primaria.
- **Clave foránea:** Indica si la columna seleccionada es clave foránea.
- **Auto incrementar:** Indica si la columna seleccionada se debe incrementar automáticamente.
- **No Nulo:** Indica si la columna seleccionada no puede tomar el valor nulo.
- **Único:** Indica si la columna seleccionada no puede tomar valores repetidos.
- **Por defecto:** Indica el valor que toma la columna por defecto. Solo mostrará y almacenará información en caso de que la casilla *Usar por defecto* este marcada.
- **Usar por defecto:** Indica si se debe utilizar un valor por defecto.
- **Título:** Nombre más descriptivo para la columna, usado en el diccionario de datos.
- **Descripción:** Descripción completa de la columna, usada en el diccionario de datos.

Toda la información que se modifique en las columnas tendrá impacto sobre el diagrama, ya sea visualmente, a la hora de generar código o en el diccionario de datos.

Panel edición



Figura E.15: Panel edición

En esta sección encontramos diferentes botones con los que editar el diagrama. Los botones disponibles son:

- **Borrar elemento:** Elimina el elemento seleccionado.
- **Borrar todo:** Elimina todos los elementos del diagrama.
- **Deshacer:** Deshace la última acción realizada en el diagrama.
- **Rehacer:** Rehace la última acción deshecha.
- **Imagen:** Muestra una pestaña nueva en el navegador con una imagen de todos los elementos que el usuario haya añadido al diagrama.
- **Exportar XML:** Muestra una ventana modal con el diagrama en código XML, permitiendo al usuario descargar el contenido en un fichero para continuar con la edición posteriormente.
- **Importar XML:** Importa un diagrama desde un fichero XML. El diagrama importado debe estar formateado correctamente.

Diccionario de datos

En este apartado encontraremos una descripción más detallada del significado de cada columna de nuestro diagrama. La información del diccionario de datos debe ser previamente introducida por el usuario, en el correspondiente apartado en el panel de propiedades.

Panel vista

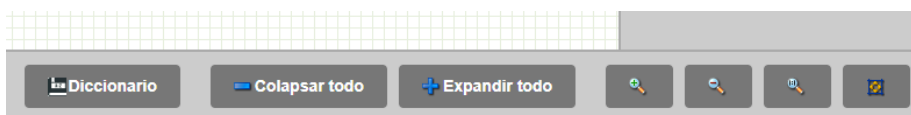


Figura E.16: Panel vista

En la sección inferior disponemos de botones que nos permiten modificar como se ve nuestro diagrama. Los botones de los que se dispone son:

- **Diccionario:** Muestra una ventana modal con el diccionario de datos correspondiente a los elementos que el usuario haya introducido en el diagrama.
- **Colapsar todo:** Colapsa todas las tablas del diagrama.
- **Expandir todo:** Expande todas las tablas del diagrama.

- **Zoom in:** Aumenta el zoom.
- **Zoom out:** Disminuye el zoom.
- **Tamaño real:** Aumenta o disminuye el zoom para que los elementos del diagrama se muestren a su tamaño real.
- **Ajustar a ventana:** Ajusta el zoom para que el tamaño del diagrama se ajuste al tamaño de la ventana.

Generación de código



Figura E.17: Generación de código

En el apartado inferior izquierdo disponemos de la sección de generación de código. Utilizando estos botones, el usuario podrá generar el código correspondiente a los elementos que haya introducido en su diagrama. La generación de código tendrá en cuenta todas las tablas incluidas en el diagrama, junto con sus columnas y relaciones, además de las propiedades especiales de cada elemento.

Se generará código a partir del diagrama únicamente para la creación de las tablas introducidas en este, el resto de configuración de la base de datos depende del usuario.

El código se mostrará en una ventana modal formada por un área de texto desde la que el usuario podrá tanto copiar como modificar el código obtenido

Hasta el momento, disponemos de las siguientes posibilidades en cuanto a generación de código.

SQL

Después de añadir diferentes elementos al diagrama, pulsando el botón *SQL* obtendremos el correspondiente código necesario para la creación de las tablas que hemos introducido.

SQLAlchemy

A partir del diagrama del usuario podemos obtener el código para crear las clases que representarán las tablas en la librería *SQLAlchemy*, librería de *Python*.

Atajos de teclado

Para agilizar el uso de la aplicación y optimizar el trabajo del usuario, se han añadido algunos atajos de teclado. Los atajos incluidos en la aplicación son los siguientes.

- **Flecha derecha:** Selecciona el siguiente elemento en el plano en el que estemos. Es decir, recorre las tablas y relaciones del diagrama, o cambia entre las columnas de dentro de una tabla.
- **Flecha izquierda:** Selecciona el elemento anterior, igual que flecha derecha.
- **Flecha abajo:** Selecciona el hijo, es decir, si tenemos seleccionada una tabla, cambia a las columnas de esta.
- **Flecha arriba:** Selecciona al padre. Igual que flecha abajo, pero esta vez selecciona la tabla correspondiente desde una columna.
- **Suprimir:** Borra del diagrama el elemento seleccionado.
- **Ctrl+A:** Selecciona todos los elementos del diagrama.
- **Ctrl+Z:** Deshacer.

- **Ctrl+Y:** Rehacer.
- **F2:** Permite cambiar la etiqueta del elemento seleccionado.
- **Num pad +:** Aumenta el zoom.
- **Num pad -:** Disminuye el zoom.
- **W:** Sube de posición una columna dentro de una tabla.
- **S:** Baja de posición una columna dentro de una tabla.
- **Borrar:** Colapsa la tabla seleccionada.
- **Intro:** Expande la tabla seleccionada.
- **F4:** Añade una columna a la tabla seleccionada.
- **Ctrl+Click derecho:** Desplaza la vista sobre el diagrama.
- **Fin:** Recarga el diagrama, en caso de que algo no haya ido bien.

Anexo de sostenibilización curricular

F.1. Introducción

La sostenibilidad es un pilar fundamental en el ámbito educativo y profesional. Si buscamos satisfacer nuestras necesidades a la vez que dejamos un mundo que puedan heredar nuestros hijos, necesitamos que una educación basada en el desarrollo sostenible, no sólo en el ámbito ecológico, sino también en el social y en el económico, se encuentre disponible para todos.

El desarrollo de este proyecto ha estado enfocado en la creación de una aplicación web que permita el diseño de diagramas relacionales utilizando la notación de patas de cuervo. Durante dicho desarrollo, se han tenido en cuenta el posible impacto que esta aplicación pudiera tener tanto en el ámbito ambiental, como en el social y económica, además de haber tenido siempre en mente su viabilidad a largo plazo, siguiendo los criterios generales para la sostenibilización curricular de la CRUE[1].

Según dichos criterios, se han desarrollado los siguientes puntos.

F.2. Sostenibilidad ambiental

En cuanto a la *sostenibilidad ambiental*, el desarrollo de una aplicación web presenta varios desafíos y oportunidades. Aunque los elementos con HTML y CSS no necesitan de muchos recursos, Javascript sí que se trata de un lenguaje más demandante. Para el despliegue de la aplicación se ha utilizado la plataforma *Vercel*, la cual cuenta con una política de energía

verde con la que pretende reducir la huella de carbono producida por las aplicaciones que alberga[5].

Durante el desarrollo se ha tratado de implementar el código más eficiente posible, tratando de reducir el número de instrucciones necesarias para ejecutar la aplicación con normalidad, reduciendo su posible impacto ambiental.

Al tratarse de una aplicación que se ejecuta desde el lado del cliente, no es necesario una comunicación constante con el servidor, lo que reduce en gran manera el consumo de recursos. Además, la posibilidad de importar y exportar los diagramas en ficheros XML favorece la reutilización de estos, evitando un uso excesivo de recursos en la aplicación realizando trabajo repetido.

F.3. Sostenibilidad social

Sobre la *sostenibilidad social*, la accesibilidad de la aplicación es el punto más importante. La aplicación es de libre acceso para todo el mundo y desde todo el mundo, sin necesidad de pagos de ningún tipo. Además, el código de la aplicación se encuentra disponible para todo el mundo en su correspondiente repositorio, fomentando la prolongación del desarrollo.

F.4. Sostenibilidad económica

La *sostenibilidad económica* en este proyecto guarda relación con el resto de puntos. Al hacer el código disponible para todo el mundo, se consigue más fácilmente un mantenimiento prolongado de la aplicación, por personas que estén dispuestas a ello. De esta forma, reducimos costes en mantenimiento y desarrollo y fomentamos que todo el mundo participe en el desarrollo.

Además, habiendo desarrollado el código pensando en su eficiencia para una sostenibilidad ambiental, consigue un menor uso de recursos, por lo que la aplicación resulta más fácil de mantener económicamente hablando.

F.5. Integración de la sostenibilidad en el currículum

Es interesante de cara a tener un currículum más completo tener un trabajo previo en los criterios de sostenibilidad de la CRUE. De esta forma,

podemos demostrar que somos capaces de, en la medida de lo posible, evaluar el impacto ambiental de nuestras aplicaciones, tener en cuenta la accesibilidad y usabilidad del usuario, conocer tecnologías sostenibles, desarrollar un proyecto de código abierto y estamos abiertos a continuar con la formación y sensibilización en este tema.

F.6. Conclusiones

El desarrollo de una aplicación web incluye tener en cuenta multitud de criterios de sostenibilidad, lo que implica una formación en la materia para poder aplicar dichos criterios de forma eficiente, mejorando la calidad del trabajo final además de ser más conscientes de la necesidad de un futuro más sostenible y equitativo.

En conclusión, la sostenibilidad implica tener un enfoque holístico que considere la eficiencia energética, la inclusión social y económica, incluyendo estos aspectos antes, durante y después del desarrollo para promover un desarrollo humano y ambientalmente sostenible.

Bibliografía

- [1] CRUE. Directrices para la introducción de la sostenibilidad en el curriculum. https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf, 2024. [Internet; descargado 8-junio-2024].
- [2] Anna He. ¿cuánto tiempo dura un ordenador portátil? <https://www.minitool.com/es/respaldar-datos/cuanto-dura-un-ordenador-portatil.html>, 2023. [Internet; descargado 28-mayo-2024].
- [3] infoautónomos. Las retenciones del irpf de los profesionales autónomos. <https://www.infoautonomos.com/fiscalidad/retenciones-irpf-profesionales-autonomos/>, 2024. [Internet; descargado 28-mayo-2024].
- [4] Seguridad Social. Bases y tipos de cotización 2024. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537>, 2024. [Internet; descargado 28-mayo-2024].
- [5] Vercel. What is vercel's green energy policy? <https://vercel.com/guides/what-is-vercel-green-energy-policy>, 2024. [Internet; descargado 8-junio-2024].
- [6] Visure. Requisitos no funcionales: tipos, ejemplos y enfoques. <https://visuresolutions.com/es/requirements-management-traceability-guide/non-functional-requirements/>, 2024. [Internet; descargado 23-mayo-2024].

- [7] Wikipedia. Just-in-time compilation. https://en.wikipedia.org/wiki/Just-in-time_compilation, 2024. [Internet; descargado 28-mayo-2024].