



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN INGENIERÍA DEL SOFTWARE

**Juez de consultas SQL
basado en Postgresql**

Alumno:
Raúl Medina González

Tutor:
Manuel Barrio Solórzano

Agradecimientos

*A mi familia, por estar ahí en mis mejores y peores momentos.
Os quiero.*

*A Yaiza, por el apoyo incondicional y darme ánimos en los momentos más importantes.
Gracias por alegrar mis días.*

*A Javier, por hacerme sentir acompañado.
Los problemas se combaten mejor sabiendo que no estás solo.*

*A Rubén y Marta, por acudir rápidamente a una llamada de emergencia.
Hacéis que cada segundo valga el doble.*

*A Manuel, por acompañarme en esta aventura.
Gracias por confiar en mi potencial.*

*A mis amigos y compañeros, por compartir sus ratos libres conmigo.
No sois conscientes de lo que podéis ayudar con vuestra mera compañía.*

Resumen

Este documento recoge el proceso de desarrollo de una aplicación de escritorio para la corrección de consultas SQL. El objetivo del proyecto es proporcionar una herramienta para el aprendizaje y la enseñanza en los fundamentos de bases de datos relacionales. La aplicación ha sido desarrollada en Python usando Postgresql como sistema gestor de bases de datos.

Este proyecto continúa desarrollando la idea de los jueces de código y abre el camino para nuevos proyectos de aprendizaje interactivo en el ámbito de las bases de datos.

Palabras clave

Juez, SQL, Python, Postgresql, bases de datos.

Abstract

This document includes the developmental process of a desktop application to SQL query correction. The goal of this project is to provide a tool to teach and learn about relational databases fundamentals. The application has been developed in Python using Postgresql as its database management system.

This project continues to expand the code judge idea and opens a new path for new projects of interactive learning in the field of databases.

Key words

Judge, SQL, Python, Postgresql, databases.

Índice general

Agradecimientos	I
Resumen	III
Abstract	V
Lista de figuras	X
Lista de tablas	XII
1. Introducción	1
1.1. Introducción	1
1.1.1. Contexto	1
1.1.2. Motivación	1
1.1.3. Objetivos	2
1.1.4. Objetivos de la aplicación	2
1.1.5. Objetivos de aprendizaje	3
1.1.6. Situación de la cuestión	3
2. Requisitos	5
2.1. Descripción de requisitos	5
2.1.1. Requisitos funcionales	5

2.1.2. Requisitos no funcionales	7
2.1.3. Requisitos de información	8
2.1.4. Restricciones	8
2.2. Casos de uso	9
2.2.1. Detalle de casos de uso	9
3. Plan de proyecto	19
3.1. Metodología	19
3.1.1. Descripción	19
3.1.2. Fases	19
3.2. Plan de trabajo	21
3.2.1. Análisis de riesgos	24
3.3. Seguimiento	26
3.4. Costes	29
4. Análisis	31
4.1. Usuarios objetivo	31
4.2. Diagrama de casos de uso	32
4.3. Modelo de dominio	35
5. Diseño	37
5.1. Arquitectura	37
5.2. Diseño detallado	38
5.3. Base de datos	47
5.4. Interfaz gráfica	48
6. Plan de pruebas	63
6.1. Diseño dirigido por pruebas	63
6.2. Pruebas unitarias	63

<i>ÍNDICE GENERAL</i>	IX
6.2.1. conftest	64
6.2.2. column_test	65
6.2.3. analyzer_test	65
6.2.4. connection_manager_test	65
6.2.5. context_test	68
6.2.6. custom_fakes_test	68
6.2.7. filler_test	70
6.2.8. judge_test	70
6.2.9. question_test	71
6.2.10. squema_recollector_test	72
6.2.11. type_compatible_test	73
6.2.12. fake_data_gen_test	74
7. Conclusiones y líneas futuras	79
7.1. Conclusión y trabajo futuro	79
7.1.1. Objetivos alcanzados	79
7.1.2. Objetivos no alcanzados	79
7.1.3. Líneas de trabajo futuro	80
7.1.4. Opinión personal	80
A. Manual de instalación	81
A.1. Requisitos	81
A.2. Instrucciones de instalación	82
B. Manual de usuario	85
B.1. Menú principal	86
B.2. Menú de escenarios disponibles	87
B.3. Resolver preguntas	88

B.4. Informe de corrección	91
B.5. Menú de administración	92
B.6. Creación de escenarios	92
B.6.1. Esquema	93
B.6.2. Datos	94
B.6.3. Preguntas	96
B.7. Edición de escenarios	98
B.7.1. Configuración general	98
B.7.2. Generar datos	99
B.7.3. Configurar preguntas	100
B.8. Importación de tipos personalizados	101
B.9. Edición de tipos personalizados	103
C. Manual de mantenimiento	105
C.1. General	106
C.2. Núcleo de la aplicación	106
C.3. Interfaz	107
Bibliografía	109

Lista de figuras

4.1. Diagrama de casos de uso relacionados con escenarios	32
4.2. Diagrama de casos de uso relacionados con respuestas	33
4.3. Diagrama de casos de uso relacionados con configuraciones	34
4.4. Modelo de dominio	35
5.1. Arquitectura propuesta	37
5.2. Diseño detallado de dbjudge	38
5.3. Diseño detallado de connection manager	39
5.4. Diseño detallado de fake data gen	40
5.5. Diseño detallado de judge	41
5.6. Diseño detallado de custom fakes	42
5.7. Diseño detallado de questions	43
5.8. Diseño detallado de structures	44
5.9. Diseño detallado de utils	45
5.10. Diseño detallado de exceptions	46
5.11. Modelo relacional	47
5.12. Estructura de vistas de la interfaz de usuario	48
5.13. Menú principal	49
5.14. Menú de administración	50
5.15. Vista de creación de escenario - Paso 1	51

5.16. Vista de creación de escenario - Paso 2	52
5.17. Vista de creación de escenario - Paso 3	53
5.18. Vista de edición de escenario - General	54
5.19. Vista de edición de escenario - Datos	55
5.20. Vista de edición de escenario - Preguntas	56
5.21. Vista para importar tipos de datos	57
5.22. Vista para editar tipos importados	58
5.23. Menú de selección de pruebas	59
5.24. Vista de resolución de preguntas	60
5.25. Vista de resultados de una prueba	61
A.1. Instalación paso 1	82
A.2. Instalación paso 2	83

Lista de tablas

2.1. Requisitos funcionales 1	5
2.2. Requisitos funcionales 2	6
2.3. Requisitos funcionales 3	7
2.4. Requisitos no funcionales	7
2.5. Requisitos de información	8
2.6. Restricciones	8
2.7. CU-P1	9
2.8. CU-P2	10
2.9. CU-P3	10
2.10. CU-P4	11
2.11. CU-E1	11
2.12. CU-E2	12
2.13. CU-E3	12
2.14. CU-E4	13
2.15. CU-E5	13
2.16. CU-E6	14
2.17. CU-E7	14
2.18. CU-E8	15
2.19. CU-E9	16

2.20. CU-E10	16
2.21. CU-E11	17
2.22. CU-E12	18
2.23. CU-C1	18
3.1. Planificación de la fase inicial	21
3.2. Planificación de la fase de elaboración	22
3.3. Planificación de la fase de construcción	22
3.4. Planificación de la fase de transición	23
3.5. Seguimiento de la fase inicial	26
3.6. Seguimiento de la fase de elaboración	27
3.7. Seguimiento de la fase de construcción	28
3.8. Seguimiento de la fase de transición	28

Capítulo 1

Introducción

1.1. Introducción

1.1.1. Contexto

En la actualidad, la tecnología ha invadido casi todos los ámbitos de las actividades cotidianas por lo que no es extraño encontrarse soluciones tecnológicas que ya se han establecido en pequeñas y medianas empresas.

Uno de estos ámbitos es la educación. El uso de las tecnologías para asistir tanto a alumnos como a profesores ha cambiado radicalmente la dinámica en las aulas durante las últimas décadas. Desde poder enseñar usando un proyector, pasando por proveer recursos digitales a través de internet, hasta el uso de videojuegos para aprovechar el potencial motivador de estos y canalizarlo en el aprendizaje.

En nuestro caso nos vamos a centrar en un tipo de aplicación interactiva que permita a los alumnos poner a prueba los conocimientos adquiridos en el aula. Ayudando a afianzar conceptos y a despertar dudas que podrían haber pasado desapercibidas sin enfrentarse a un caso práctico. Concretamente, en el ámbito de la enseñanza sobre fundamentos de bases de datos, una de las asignaturas que se estudian en la carrera de ingeniería informática.

1.1.2. Motivación

Una de las grandes luchas de la educación en los últimos tiempos, sobre todo en el ámbito de las ingenierías, ha sido armonizar la parte teórica de una asignatura con su parte práctica. Aprovechando la forma de aprender del ser humano se pueden reforzar los fundamentos teóricos si estos son aplicados en casos reales.

Una de las formas de aprovecharse de cómo funciona nuestro cerebro a la hora de aprender es la gamificación. La gamificación es la capacidad de transferir las propiedades de un videojuego a otro ámbito que no sea puramente lúdico. Las aplicaciones de aprendizaje intentan aprovechar las diferentes propiedades de los videojuegos para hacer que el proceso didáctico sea más sencillo, efectivo y agradable.

Uno de los tipos de aplicaciones que ponen la gamificación en práctica son los jueces de código. Los jueces de código plantean problemas a resolver y comparan los resultados de la solución propuesta por el usuario con los resultados de una solución correcta para ese problema. Después de comparar los resultados y hacer otros análisis automáticos adicionales, el sistema devuelve un veredicto. En el resultado, el alumno puede ver un pequeño análisis de su solución. Este informe puede contener una recopilación de requisitos obligatorios y también otros datos relevantes que no afectan al resultado final.

De esta manera, el usuario puede saber si ha acertado o no en su respuesta y puede incluso acceder a una lista de posibles errores, pero sin llegar a ver la solución correcta en caso de no haberlo conseguido. Esto fomenta el pensamiento crítico y la resolución de problemas. En caso de disponer de algún tipo de puntuación (basada, por ejemplo, en el tiempo de ejecución, el espacio consumido o el número de líneas), la aplicación podría proporcionar también una fuente de motivación personal. Esto puede provocar también una motivación competitiva si se comparten los mejores resultados en línea.

Este proyecto intenta emular a los jueces de código, aplicando los mismos principios para el aprendizaje de SQL *Standard Query Language*, el lenguaje más usado para comunicarse con las bases de datos relacionales. Este trabajo de fin de grado pretende hacer avanzar un paso más el uso de las tecnologías en la enseñanza y quiere servir de inspiración para continuar cambiando las aulas.

1.1.3. Objetivos

El objetivo principal de este proyecto es conseguir una aplicación de escritorio que permita juzgar consultas SQL respecto a unas preguntas planteadas sobre un dominio específico correspondiente a una base de datos.

1.1.4. Objetivos de la aplicación

Se espera poder alcanzar al menos los siguientes objetivos generales:

- Rellenado de bases de datos con datos generados aleatoriamente.
- Comprobación de resultados entre consultas equivalentes.
- Comprobación de tablas usadas.
- Comprobación de palabras clave.

- Generación de datos adicionales generados específicamente para una consulta correcta para aumentar la precisión del juez.
- Opciones de generación de datos personalizadas.

1.1.5. Objetivos de aprendizaje

Después de la realización del proyecto se espera tener un mejor conocimiento en las siguientes áreas:

- Aprendizaje del framework Psycopg2 de gestión de conexiones con bases de datos.
- Aprendizaje del framework PyQt5 de creación de interfaces gráficas.
- Consolidación y profundización del estándar SQL-92 visto durante la carrera.
- Consolidación y profundización en el uso del sistema gestor de bases de datos Postgresql.
- Preparación para el mundo laboral.

1.1.6. Situación de la cuestión

Los jueces de código son bastante populares sobre todo debido a la programación competitiva que requiere de práctica constante. Estos jueces proporcionan un entorno en el que pulir las habilidades de programación y continuar aprendiendo mientras te enfrentas a nuevos desafíos y problemas.

En el caso concreto del aprendizaje del lenguaje SQL, existe la página web de [sql-ex\[1\]](#). Una página disponible en ruso e inglés que ofrece un aprendizaje sobre el estándar SQL-92 aunque con alguna restricción como la carencia del “NATURAL JOIN” entre tablas. Esta es la aplicación más parecida a los objetivos de este proyecto. Sin conocer los detalles de su implementación, la generación de datos podría no estar presente. Es común en este tipo de aplicaciones que las bases de datos se creen y rellenen de forma manual.

Otros enfoques sobre el aprendizaje de este área son:

- Entorno libre de pruebas en [Sqlfiddle\[2\]](#).
- Tutoriales interactivos en [Livesql\[3\]](#) proporcionados por oracle.
- Un juez en formato de retos individuales combinado con el formato de wiki en [SQLZoo\[4\]](#).
- Un videojuego que usa sentencias SQL para dirigir una flota espacial ficticia en [Schemaverse\[5\]](#).
- Un libro de texto interactivo en [Sql-tutorial\[6\]](#).
- La famosa plataforma de aprendizaje interactivo [W3schools\[7\]](#).

Capítulo 2

Requisitos

2.1. Descripción de requisitos

2.1.1. Requisitos funcionales

Identificador	Prioridad	Descripción
RF1	ALTA	El sistema deberá poder crear una base de datos a partir de un archivo DDL con su definición.
RF2	ALTA	El sistema podrá generar datos aleatoriamente.
RF2.1	ALTA	El sistema podrá generar aleatoriamente números enteros.
RF2.2	ALTA	El sistema podrá generar aleatoriamente números decimales.
RF2.3	ALTA	El sistema podrá generar aleatoriamente fechas.
RF2.4	ALTA	El sistema podrá generar aleatoriamente cadenas de caracteres.
RF2.5	ALTA	El sistema podrá generar aleatoriamente valores booleanos.
RF2.6	ALTA	El sistema podrá generar tuplas que referencien a datos de otras tablas.

Tabla 2.1: Requisitos funcionales 1

RF3	ALTA	El sistema podrá generar datos aleatoriamente entre unos límites inferiores y superiores especificados por el usuario.
RF3.1	ALTA	El sistema podrá generar aleatoriamente números enteros entre unos límites inferiores y superiores especificados por el usuario.
RF3.2	ALTA	El sistema podrá generar aleatoriamente números decimales entre unos límites inferiores y superiores especificados por el usuario.
RF3.3	ALTA	El sistema podrá generar aleatoriamente fechas entre unos límites inferiores y superiores especificados por el usuario.
RF4	ALTA	El sistema podrá generar datos aleatorios a partir de expresiones regulares.
RF5	ALTA	El sistema podrá generar datos aleatorios a partir de datos cargados previamente.
RF6	ALTA	El sistema permitirá cargar conjuntos de datos personalizados.
RF6.1	ALTA	El sistema permitirá cargar conjuntos de datos personalizados de forma manual.
RF6.2	ALTA	El sistema permitirá cargar conjuntos de datos personalizados a partir de un archivo CSV.
RF7	ALTA	El sistema permitirá editar los conjuntos de datos personalizados.
RF8	ALTA	El sistema permitirá eliminar conjuntos de datos personalizados.
RF9	ALTA	El sistema permitirá visualizar la estructura de tablas y columnas de aquellas bases de datos que se hayan creado.
RF10	ALTA	El sistema podrá rellenar las bases de datos creadas, con tuplas de datos generados aleatoriamente.
RF11	ALTA	El sistema permitirá crear escenarios que agrupen un conjunto de preguntas asociadas a una base de datos creada.
RF12	ALTA	El sistema podrá crear datos adicionales para intentar aumentar la precisión de las correcciones.
RF13	ALTA	El sistema permitirá editar las preguntas de un escenario.
RF14	ALTA	El sistema permitirá eliminar preguntas de un escenario.
RF15	ALTA	El sistema permitirá comprobar los resultados de las respuestas correctas con los resultados de las respuestas del usuario.
RF16	ALTA	El sistema permitirá probar las respuestas del usuario mostrando su resultado.
RF17	ALTA	El sistema permitirá eliminar un escenario completamente, con sus preguntas y su base de datos eliminadas completamente.
RF18	ALTA	El sistema permitirá eliminar los datos de una base de datos perteneciente a un escenario.
RF19	ALTA	El sistema permitirá generar aleatoriamente números decimales con una cantidad específica de posiciones decimales.

Tabla 2.2: Requisitos funcionales 2

RF20	ALTA	El sistema permitirá añadir palabras clave obligatorias para una pregunta.
RF21	ALTA	El sistema permitirá añadir palabras clave prohibidas para una pregunta.
RF22	ALTA	El sistema permitirá editar las palabras clave asociadas a una pregunta.
RF23	ALTA	El sistema permitirá eliminar las palabras clave asociadas a una pregunta.
RF24	ALTA	El sistema podrá analizar las tablas innecesarias usadas en una respuesta.
RF25	ALTA	El sistema podrá analizar las palabras clave prohibidas usadas en una respuesta.
RF26	ALTA	El sistema podrá analizar las palabras clave obligatorias usadas en una respuesta.
RF27	ALTA	El sistema podrá analizar si la pregunta se ha dejado sin contestar.
RF27	ALTA	El sistema permitirá generar una cantidad específica de datos aleatorios para cada tabla.

Tabla 2.3: Requisitos funcionales 3

2.1.2. Requisitos no funcionales

Identificador	Prioridad	Descripción
RNF1	ALTA	El sistema no permitirá al usuario responder a una pregunta con la conexión de Postgresql en modo escritura.
RNF2	ALTA	La comparación de resultados no tendrá en cuenta el orden a no ser que se especifique con las palabras claves ORDER BY.
RNF3	ALTA	El sistema podrá funcionar al menos con Postgresql como sistema gestor de bases de datos.
RNF4	ALTA	El sistema permitirá interrumpir la generación de datos en cualquier momento.
RNF5	ALTA	El sistema creará las configuraciones necesarias para su funcionamiento si no están presentes.
RNF6	ALTA	El sistema deberá funcionar al menos en el sistema operativo Ubuntu en su versión 16.04.
RNF7	MEDIA	El sistema permitirá al usuario contestar las preguntas en cualquier orden.
RNF8	BAJA	El sistema será fácil de distribuir mediante el uso de un ejecutable.

Tabla 2.4: Requisitos no funcionales

2.1.3. Requisitos de información

Identificador	Prioridad	Descripción
R1	ALTA	Las tuplas generadas deberán respetar la integridad referencial.
R2	ALTA	El sistema deberá guardar las preguntas y respuestas de un escenario.
R3	ALTA	El sistema deberá guardar la base de datos correspondiente a cada escenario.
R4	ALTA	El sistema deberá proveer una lista de palabras clave comunes cargada en el sistema.

Tabla 2.5: Requisitos de información

2.1.4. Restricciones

Identificador	Prioridad	Descripción
R5	ALTA	La conexión utilizada con Postgresql se debe hacer con un usuario con permisos elevados.
R6	ALTA	La conexión utilizada con Postgresql debe tener unas credenciales correctas.
R7	ALTA	La versión mínima soportada de Postgresql es 9.5.
R7	ALTA	La configuración de la generación de datos no puede exigir estados imposibles.

Tabla 2.6: Restricciones

2.2. Casos de uso

Identificador	Prioridad	Descripción
CU-P1	ALTA	Obtener información sobre un escenario.
CU-P2	ALTA	Contestar pregunta.
CU-P3	ALTA	Probar una respuesta.
CU-P4	ALTA	Corregir preguntas.
CU-E1	ALTA	Configurar esquema de un escenario.
CU-E2	ALTA	Configurar generación de datos.
CU-E3	ALTA	Configurar palabras clave de una pregunta.
CU-E4	ALTA	Configurar preguntas.
CU-E5	ALTA	Crear un escenario.
CU-E6	ALTA	Borrar datos de una escenario.
CU-E7	ALTA	Eliminar escenario.
CU-E8	ALTA	Generar datos para un escenario existente.
CU-E9	ALTA	Editar preguntas de un escenario existente.
CU-E10	ALTA	Importar tipos de datos personalizados.
CU-E11	ALTA	Editar tipos de datos personalizados.
CU-E12	ALTA	Eliminar tipos de datos personalizados.
CU-C1	ALTA	Especificación de conexión con la base de datos principal.

2.2.1. Detalle de casos de uso

Identificador	CU-P1	
Título	Obtener información sobre un escenario	
Precondición	Hay escenarios cargados en el sistema.	
Pasos	Paso	Descripción
	1	El actor 'usuario' solicita ver la información de un escenario.
	2	El sistema muestra el número de tablas en la base de datos del escenario.
	3	El sistema muestra el número de preguntas del escenario.
4	El sistema muestra el número de tuplas existentes en la base de datos del escenario.	
Postcondición	El actor 'usuario' verá un resumen del contenido del escenario.	
Excepciones		

Tabla 2.7: CU-P1

Identificador	CU-P2	
Título	Contestar pregunta	
Precondición	Hay escenarios cargados en el sistema. El actor ‘usuario’ se encuentra dentro de un escenario, preparado para contestar preguntas.	
Pasos	Paso	Descripción
	1	El actor ‘usuario’ selecciona un escenario para contestar preguntas.
	2	El sistema muestra las preguntas del escenario.
	3	El actor ‘usuario’ selecciona una pregunta.
	4	El sistema muestra el enunciado de la pregunta.
	5	El actor ‘usuario’ introduce una respuesta.
6	El sistema almacena la respuesta contestada y queda pendiente de corrección.	
Postcondición	La respuesta del ‘usuario’ se almacenará temporalmente y quedará pendiente de corrección.	
Excepciones	Paso	Descripción
	3a	El actor ‘usuario’ decide no continuar y el caso de uso queda sin efecto.
	5a	El actor ‘usuario’ decide no continuar y el caso de uso queda sin efecto.

Tabla 2.8: CU-P2

Identificador	CU-P3	
Título	Probar una respuesta	
Precondición	Se acaba de completar el caso de uso CU-P2.	
Pasos	Paso	Descripción
	1	El actor ‘usuario’ selecciona <i>Probar consulta</i> .
	2	El sistema muestra el resultado de ejecutar la consulta respuesta en la base de datos del escenario.
Postcondición	Se muestra el resultado de la respuesta.	
Excepciones	Paso	Descripción
	2a	Si la consulta está mal construida o no puede ejecutarse en la base de datos, el sistema mostrará un aviso informando de ello.

Tabla 2.9: CU-P3

Identificador	CU-P4	
Título	Corregir preguntas	
Precondición	Se ha ejecutado el caso de uso CU-P2.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona <i>Terminar y corregir preguntas</i> .
	2	El sistema genera un informe para cada pregunta.
	3	El sistema muestra el número de preguntas acertadas del total.
	4	El sistema muestra las tablas innecesarias usadas en cada respuesta.
	5	El sistema muestra si el resultado de cada consulta es correcto.
	6	El sistema muestra si cada respuesta se considera correcta.
	7	El sistema muestra para cada pregunta las palabras clave prohibidas y si se han utilizado.
8	El sistema muestra para cada pregunta las palabras clave obligatorias y si se han utilizado.	
Postcondición	Se muestra un informe a partir de las preguntas del escenario que se hayan contestado.	
Excepciones		

Tabla 2.10: CU-P4

Identificador	CU-E1	
Título	Configurar esquema de un escenario	
Precondición	El actor 'usuario' se encuentra en el menú de creación de escenarios.	
Pasos	Paso	Descripción
	1	El actor 'usuario' introduce un nombre sin usar para el escenario.
	2	El actor 'usuario' introduce una definición de tablas de base de datos.
	3	El actor 'usuario' confirma la definición introducida.
	4	El sistema crea un nuevo escenario y una nueva base de datos asociada.
5	El sistema habilita la navegación hacia la configuración de la generación de datos aleatorios.	
Postcondición	Se crea un nuevo escenario sin preguntas y una nueva base de datos asociada al escenario.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	3a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.11: CU-E1

Identificador	CU-E2	
Título	Configurar generación de datos	
Precondición	Se ha ejecutado correctamente el CU-E1. El actor 'usuario' se encuentra en la ventana de generación de datos.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona una tabla.
	2	El sistema muestra la lista de columnas de la tabla seleccionada.
	3	El actor 'usuario' cambia la configuración por defecto de una columna.
	4	El actor 'usuario' navega hacia la configuración de preguntas.
	5	El sistema guarda la configuración de generación de datos.
Postcondición		
Excepciones	Paso	Descripción
	3a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4b	El actor 'usuario' decide modificar otra tabla, el caso de uso continúa en el paso 1.
	4b	El actor 'usuario' decide modificar otra columna, el caso de uso continúa en el paso 3.

Tabla 2.12: CU-E2

Identificador	CU-E3	
Título	Configurar palabras clave de una pregunta	
Precondición	El actor 'usuario' ha seleccionado <i>Palabras clave</i> en una pregunta y se encuentra en la ventana de modificación de palabras clave.	
Pasos	Paso	Descripción
	1	El actor 'usuario' escoge prohibir o obligar la palabra clave.
	2	El actor 'usuario' escoge la palabra clave.
	3	El sistema muestra la palabra clave con su restricción, añadida a la lista de palabras claves de la pregunta.
	4	El actor 'usuario' confirma las palabras clave añadidas.
	5	El sistema almacena las palabras clave asociadas a la pregunta seleccionada.
Postcondición	Las palabras clave añadidas quedan asociadas a la pregunta.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.13: CU-E3

Identificador	CU-E4	
Título	Configurar preguntas	
Precondición	El actor 'usuario' se encuentra en la ventana de configuración de preguntas.	
Pasos	Paso	Descripción
	1	El actor 'usuario' introduce el enunciado de una pregunta.
	2	El actor 'usuario' introduce una consulta correcta como respuesta a la pregunta.
	3	El actor 'usuario' selecciona <i>Añadir pregunta</i> .
	4	El sistema añade la pregunta y la respuesta a la lista de preguntas.
5	El actor decide que ha terminado de configurar preguntas y el caso de uso termina.	
Postcondición	Las preguntas y respuestas quedan temporalmente almacenadas a la espera de terminar la creación del escenario.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	3a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	5a	El actor 'usuario' decide configurar otra pregunta y el caso de uso continúa en el paso 1.

Tabla 2.14: CU-E4

Identificador	CU-E5	
Título	Crear un escenario	
Precondición	El actor 'usuario' ha configurado un escenario siguiendo los casos de uso CU-E1, CU-E2 y CU-E3.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona <i>Terminar</i> .
	2	El sistema crea un escenario completo.
Postcondición	Se almacena de forma persistente un escenario con sus preguntas y los datos generados en su base de datos.	
Excepciones		

Tabla 2.15: CU-E5

Identificador	CU-E6	
Título	Borrar datos de un escenario	
Precondición	Existe al menos un escenario. El actor 'usuario' se encuentra en la pantalla de edición general de escenarios.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona un escenario.
	2	El actor 'usuario' selecciona <i>Borrar datos del escenario</i> .
	3	El sistema elimina todos los datos de la base de datos asociada al escenario.
Postcondición	Los datos de la base de datos asociada al escenario quedan eliminados.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.16: CU-E6

Identificador	CU-E7	
Título	Eliminar escenario	
Precondición	Existe al menos un escenario. El actor 'usuario' se encuentra en la pantalla de edición general de escenarios.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona un escenario.
	2	El actor 'usuario' selecciona <i>Eliminar escenario</i> .
	3	El sistema elimina el escenario y toda su información asociada.
Postcondición	El escenario y toda su información son eliminados.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.17: CU-E7

Identificador	CU-E8	
Título	Generar datos para un escenario existente	
Precondición	Existe al menos un escenario. El actor 'usuario' se encuentra en la pantalla de edición de datos.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona un escenario.
	2	El actor 'usuario' selecciona una tabla.
	3	El sistema muestra la lista de columnas de la tabla seleccionada.
	4	El actor 'usuario' cambia la configuración por defecto de una columna.
	5	El actor 'usuario' decide que ha terminado de configurar los datos.
	6	El actor 'usuario' selecciona <i>Generar datos</i> .
7	El sistema genera datos en la base de datos seleccionada de acuerdo a la configuración establecida.	
Postcondición	Se han generado datos en la base de datos asociada al escenario.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	5a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	5b	El actor 'usuario' decide modificar otra tabla, el caso de uso continúa en el paso 2.
5b	El actor 'usuario' decide modificar otra columna, el caso de uso continúa en el paso 4.	

Tabla 2.18: CU-E8

Identificador	CU-E9	
Título	Editar preguntas de un escenario existente	
Precondición	Existe al menos un escenario con preguntas creadas. El actor 'usuario' se encuentra en la pantalla de edición de preguntas.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona una pregunta.
	2	El actor 'usuario' selecciona <i>Borrar</i> .
	3	El sistema elimina la pregunta de la lista.
	4	El actor 'usuario' añade de nuevo la pregunta con la modificación deseada.
	5	El sistema muestra la pregunta en la lista de nuevo.
	6	El actor 'usuario' selecciona <i>Guardar preguntas</i> .
7	El sistema guarda las preguntas en persistencia.	
Postcondición	Las preguntas del escenario son modificadas y guardadas de nuevo en persistencia.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	6a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.19: CU-E9

Identificador	CU-E10	
Título	Importar tipos de datos personalizados	
Precondición	El actor 'usuario' se encuentra en la pantalla de importación de tipos.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona <i>Cargar tipos desde archivo</i> .
	2	El actor 'usuario' selecciona un archivo CSV con los tipos deseados.
	3	El sistema muestra los tipos cargados.
	4	El actor 'usuario' selecciona <i>Guardar tipos</i> .
5	El sistema guarda los tipos en almacenamiento persistente.	
Postcondición	Se han cargado nuevos tipos personalizados en el sistema.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.20: CU-E10

Identificador	CU-E11	
Título	Editar tipos de datos personalizados	
Precondición	Existe al menos un tipo de dato personalizado. El actor 'usuario' se encuentra en la pantalla de edición de tipos personalizados.	
Pasos	Paso	Descripción
	1	El actor 'usuario' hace doble click en el dato a modificar.
	2	El actor 'usuario' introduce un dato diferente al anterior.
	3	El actor 'usuario' aprieta la tecla <i>Enter</i> .
	4	El sistema muestra en pantalla los cambios realizados.
	5	El actor 'usuario' selecciona <i>Guardar tipos</i> .
6	El sistema guarda los tipos en persistencia.	
Postcondición	Se han modificado los tipos personalizados en el sistema.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	3a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	5a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	5b	El actor 'usuario' decide editar otro dato y el caso continúa en el paso 1.

Tabla 2.21: CU-E11

Identificador	CU-E12	
Título	Eliminar tipos de datos personalizados	
Precondición	Existe al menos un tipo de dato personalizado. El actor 'usuario' se encuentra en la pantalla de edición de tipos personalizados.	
Pasos	Paso	Descripción
	1	El actor 'usuario' selecciona el tipo de dato a eliminar.
	2	El actor 'usuario' selecciona <i>Eliminar selección</i> .
	3	El sistema elimina ese tipo de dato con sus datos asociados de la lista de tipos personalizados.
	4	El actor 'usuario' selecciona <i>Guardar tipos</i> .
	5	El sistema guarda los cambios en persistencia.
Postcondición	Se han eliminado los tipos de datos deseados.	
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.
	4b	El actor 'usuario' decide eliminar más tipos y el caso de uso continúa en el paso 1.

Tabla 2.22: CU-E12

Identificador	CU-C1	
Título	Especificación de conexión con la base de datos principal	
Precondición	La conexión con la base de datos no está configurada. El sistema se ha iniciado correctamente.	
Pasos	Paso	Descripción
	1	El actor 'usuario' introduce las credenciales de la conexión.
	2	El actor 'usuario' selecciona <i>Guardar</i> .
	3	El sistema guarda la configuración.
Postcondición		
Excepciones	Paso	Descripción
	2a	El actor 'usuario' decide no continuar y el caso de uso queda sin efecto.

Tabla 2.23: CU-C1

Capítulo 3

Plan de proyecto

3.1. Metodología

3.1.1. Descripción

Para el desarrollo de este proyecto se ha utilizado el proceso unificado como marco de trabajo. Este marco de trabajo extensible es conocido por ser utilizado por la empresa IBM. Las principales características del proceso unificado son:

- Estar dirigido por casos de uso.
- Estar centrado en la arquitectura.
- Ser iterativo e incremental.

3.1.2. Fases

El proceso unificado divide el proyecto en 4 fases: Comienzo, elaboración, construcción y transición. Cada una de las fases tiene varias iteraciones y al finalizar cada fase se obtienen diversos artefactos.

Para este proyecto no se producirán artefactos formales durante el desarrollo de las fases, sino que se realizarán únicamente los artefactos finales en la fase de elaboración reflejando el análisis y diseño de todo el sistema. El análisis y diseño del proyecto se hará en forma de esbozos para tener una capacidad de adaptación e iteración más rápida con el cliente.

Las iteraciones dentro de cada fase no tendrán una cantidad de tiempo fija asignada debido a las restricciones horarias en las que se desarrollará el proyecto.

Comienzo

En esta fase se fijan los objetivos del proyecto y los plazos para realizarlo. Se realiza un análisis para identificar el alcance del proyecto, los posibles riesgos que puedan surgir durante el mismo y estimaciones de costes, esfuerzo y tiempo, así como de la calidad general del proyecto. También es en esta fase cuando se comienzan a barajar posibles arquitecturas para desarrollar la solución.

El hito más importante de esta fase ocurre cuando se llega a un acuerdo en el alcance del proyecto, la arquitectura tiene en cuenta los requisitos generales más importantes y se tienen razones comerciales suficientes para comenzar con el desarrollo.

Elaboración

En esta fase se refina la dirección del proyecto. Se asientan las bases sobre las que se empezará a construir el producto final teniendo en cuenta las limitaciones de tiempo, coste y otra índole que se hayan detectado. Se recogen la mayoría de los requisitos funcionales y se refina la arquitectura de cara a la fase de construcción.

En este punto se empieza a tomar nota de los riesgos, expandiendo la lista de los identificados en el comienzo con aquellos riesgos imprevistos, así como un refinamiento de las medidas de contención para amortiguar el impacto de los mismos. También se realiza un plan detallado para guiar la fase de construcción.

El hito más importante de esta fase ocurre cuando los requisitos funcionales están incluidos en el modelo de casos de uso, la arquitectura tiene una base sólida sobre la que empezar a construir y se tiene un plan para el desarrollo del producto en la etapa de construcción.

Construcción

El objetivo principal de la fase de construcción es generar un sistema capaz de comportarse correctamente en un escenario de pruebas con clientes.

Durante esta fase se desarrollan de forma incremental e iterativa las tareas necesarias para la realización del sistema. También se comprueba la viabilidad del proyecto de manera directa y de forma continuada hasta la etapa de transición.

El hito más importante de esta fase ocurre cuando se tiene un sistema prácticamente operativo que pueda ser utilizado por el cliente en un entorno de prueba.

Transición

Esta fase se centra en terminar de pulir el sistema producido y entregarlo al cliente.

Durante esta fase se corrigen errores que hubieran pasado desapercibidos y se añaden pequeñas modificaciones para solventar problemas que no se hubieran detectado antes.

El hito más importante de esta etapa es el lanzamiento de la aplicación, y se caracteriza por la finalización del desarrollo y la distribución del sistema.

3.2. Plan de trabajo

En las siguientes tablas se pueden ver las estimaciones temporales y división de tareas de cada una de las fases. La cantidad de horas por día será flexible sin una relación directa entre días transcurridos y horas invertidas en el proyecto.

Tarea	Horas	Inicio	Fin
TOTAL	48	1/2/2019	15/2/2019
Inicio de fase	0	1/2/2019	1/2/2019
Alcance del proyecto	5	1/2/2019	5/2/2019
Definición de requisitos	2	1/2/2019	5/2/2019
Estimación de riesgos	2	1/2/2019	4/2/2019
Estimación de costes	1	1/2/2019	4/2/2019
Planificación temporal	3	1/2/2019	6/2/2019
Prueba de concepto	10	5/2/2019	8/2/2019
Arquitectura inicial	5	5/2/2019	15/2/2019
Formación	5	5/2/2019	15/2/2019
Investigación SQL-92	10	1/2/2019	15/2/2019
Entorno de desarrollo	5	5/2/2019	8/2/2019
Fin de fase	0	15/2/2019	15/2/2019

Tabla 3.1: Planificación de la fase inicial

Tarea	Horas	Inicio	Fin
TOTAL	62	15/2/2019	28/2/2019
Inicio de fase	0	15/2/2019	15/2/2019
Definición completa de requisitos	2	15/2/2019	16/2/2019
Análisis de casos de uso	2	15/2/2019	18/2/2019
Análisis del modelo de dominio	7	15/2/2019	18/2/2019
Diseño de casos de uso	3	18/2/2019	19/2/2019
Diseño del modelo de dominio	8	18/2/2019	28/2/2019
Diseño de la base de datos	5	15/2/2019	22/2/2019
Formación sobre Psycopg2	10	18/2/2019	20/2/2019
Formación sobre Postgresql	15	20/2/2019	22/2/2019
Formación sobre xeger	1	22/2/2019	25/2/2019
Formación sobre sqlparse	3	25/2/2019	26/2/2019
Formación sobre pytest	3	26/2/2019	27/2/2019
Investigación sobre la generación de datos	5	27/2/2019	28/2/2019
Fin de fase	0	28/2/2019	28/2/2019

Tabla 3.2: Planificación de la fase de elaboración

Tarea	Horas	Inicio	Fin
TOTAL	235	28/2/2019	27/3/2019
Inicio de fase	0	28/2/2019	28/2/2019
Implementación del núcleo de la aplicación de acuerdo a los requisitos	100	28/2/2019	18/2/2019
Implementación básica de la interfaz de la aplicación	50	18/2/2019	6/3/2019
Implementación de casos de uso relacionados con preguntas	25	15/3/2019	22/3/2019
Implementación de casos de uso relacionados con escenarios	50	20/2/2019	15/3/2019
Implementación de casos de uso relacionados con configuraciones	10	22/3/2019	27/3/2019
Fin de fase	0	27/3/2019	27/3/2019

Tabla 3.3: Planificación de la fase de construcción

Tarea	Horas	Inicio	Fin
TOTAL	10	27/3/2019	8/4/2019
Inicio de fase	0	27/3/2019	27/3/2019
Elaboración del manual de usuario	3	27/3/2019	8/4/2019
Elaboración del manual de instalación	2	27/3/2019	8/4/2019
Elaboración del manual de mantenimiento	5	27/3/2019	8/4/2019
Fin de fase	0	8/4/2019	8/4/2019

Tabla 3.4: Planificación de la fase de transición

3.2.1. Análisis de riesgos

A continuación se enumeran los riesgos previstos del proyecto.

Identificador	RK-1
Descripción	Implementación de funcionalidad errónea.
Prioridad	MEDIA
Medidas	Asegurar un análisis de requisitos en profundidad. Realizar iteraciones frecuentes con el cliente.

Identificador	RK-2
Descripción	Implementación de interfaz errónea.
Prioridad	MEDIA
Medidas	Realizar varios bocetos de interfaz e iteraciones múltiples con el cliente.

Identificador	RK-3
Descripción	Imposibilidad de comunicación con el cliente.
Prioridad	ALTA
Medidas	Adelantarse a los periodos en los que una de las partes no pueda reunirse obteniendo información suficiente para poder seguir trabajando.

Identificador	RK-4
Descripción	Pérdida del código fuente de la aplicación.
Prioridad	ALTA
Medidas	Tener una copia de seguridad y usar un sistema de control de versiones.

Identificador	RK-5
Descripción	Pérdida del material de trabajo.
Prioridad	BAJA
Medidas	Preparar material de repuesto con antelación.

Identificador	RK-6
Descripción	Pérdida de conexión a internet.
Prioridad	MEDIA
Medidas	Disponer de un dispositivo de conexión a internet alternativo y guardar localmente el material de consulta.

Identificador	RK-7
Descripción	Tiempo de formación insuficiente.
Prioridad	ALTA
Medidas	Realizar un solapamiento de tareas y preparar un espacio temporal de margen.

Identificador	RK-8
Descripción	Tiempo estimado para tareas insuficiente.
Prioridad	ALTA
Medidas	Realizar un solapamiento de tareas y preparar un espacio temporal de margen.

3.3. Seguimiento

En general, las estimaciones del proyecto han demostrado ser excesivamente optimistas y los tiempos se han alejado considerablemente de su fecha estimada. Las estimaciones de horas dedicadas han estado más acertadas, pero también se ha pecado de optimismo y en la gran mayoría de tareas se ha superado la estimación. Se ha conseguido alcanzar el plazo límite por muy poco, antes de octubre de 2020.

Debido a un riesgo imprevisto descrito más adelante, parte del conteo de horas será aproximado. La herramienta que se usaba para el conteo de horas tuvo un comportamiento inesperado y las horas se han aproximado usando fechas y tiempo recogidos por otras herramientas.

La fase inicial del proyecto tuvo un ligero desfase respecto a la estimación inicial. Esto fue debido a una necesidad de investigación sobre el estándar SQL-92 que llevó más tiempo del esperado. También la producción de un prototipo para comprobar la viabilidad del proyecto llevó más tiempo del planificado y requirió, a su vez, de un comienzo temprano de la siguiente fase para amortiguar el impacto de este desajuste.

Tarea	Horas estimadas	Horas reales	Inicio estimado	Fin estimado	Inicio real	Fin real
TOTAL	48	101	1/2/2019	15/2/2019	1/2/2019	29/5/2019
Inicio de fase	0	0	1/2/2019	1/2/2019	1/2/2019	29/5/2019
Alcance del proyecto	5	10	1/2/2019	5/2/2019	1/2/2019	27/2/2019
Definición de requisitos	2	5	1/2/2019	5/2/2019	1/2/2019	13/3/2019
Estimación de riesgos	2	2	1/2/2019	4/2/2019	1/2/2019	4/2/2019
Estimación de costes	1	1	1/2/2019	4/2/2019	1/2/2019	4/2/2019
Planificación temporal	3	3	1/2/2019	6/2/2019	1/2/2019	6/2/2019
Prueba de concepto	10	30	5/2/2019	8/2/2019	13/3/2019	29/5/2019
Arquitectura inicial	5	10	5/2/2019	15/2/2019	5/2/2019	13/3/2019
Formación	5	10	5/2/2019	15/2/2019	5/2/2019	29/5/2019
Investigación SQL-92	10	20	1/2/2019	15/2/2019	1/2/2019	13/3/2019
Entorno de desarrollo	5	10	5/2/2019	8/2/2019	5/2/2019	14/2/2019
Fin de fase	0	0	15/2/2019	15/2/2019	29/5/2019	29/5/2019

Tabla 3.5: Seguimiento de la fase inicial

La fase de elaboración se alargó debido a un problema originado en la fase inicial. La viabilidad de parte del proyecto era desconocida, lo cual dificultó la especificación completa de requisitos y alargó el tiempo necesario para el análisis y el diseño de las diferentes partes de la aplicación. A la mitad de esta fase se detectó un riesgo inesperado que afectaba a la planificación del proyecto. El sistema de conteo de horas usado se ha corrompido y las horas serán estimadas a partir de fechas proporcionadas en el sistema de control de versiones y fechas de reuniones con el cliente.

Se añade este riesgo a la base de riesgos para tenerlo en cuenta en futuros proyectos. Podría

haber sido evitado guardando informes periódicos con los tiempos dedicados al proyecto.

Tarea	Horas estimadas	Horas reales	Inicio estimado	Fin estimado	Inicio real	Fin real
TOTAL	62	115	15/2/2019	28/2/2019	13/3/2019	14/2/2020
Inicio de fase	0	0	15/2/2019	15/2/2019	13/3/2019	13/3/2019
Definición completa de requisitos	2	10	15/2/2019	16/2/2019	29/5/2019	14/2/2020
Análisis de casos de uso	2	10	15/2/2019	18/2/2019	29/5/2019	14/2/2020
Análisis del modelo de dominio	7	10	15/2/2019	18/2/2019	29/5/2019	14/2/2020
Diseño de casos de uso	3	13	18/2/2019	19/2/2019	29/5/2019	14/2/2020
Diseño del modelo de dominio	8	13	18/2/2019	28/2/2019	29/5/2019	14/2/2020
Diseño de la base de datos	5	3	15/2/2019	22/2/2019	29/5/2019	14/2/2020
Formación sobre Psycopg2	10	7	18/2/2019	20/2/2019	13/3/2019	27/3/2019
Formación sobre Postgresql	15	20	20/2/2019	22/2/2019	13/3/2019	16/4/2019
Formación sobre xeger	1	1	22/2/2019	25/2/2019	16/4/2019	29/5/2019
Formación sobre sqlparse	3	5	25/2/2019	26/2/2019	1/5/2019	29/5/2019
Formación sobre pytest	3	3	26/2/2019	27/2/2019	26/4/2019	30/4/2019
Investigación sobre la generación de datos	5	20	27/2/2019	28/2/2019	16/4/2019	29/5/2019
Fin de fase	0	0	28/2/2019	28/2/2019	14/2/2020	14/2/2020

Tabla 3.6: Seguimiento de la fase de elaboración

La fase de construcción ha sido la más afectada por los problemas de las fases anteriores. El solapamiento de tareas ha amortiguado gran parte de los efectos pero no ha podido evitar alejarse completamente de las estimaciones temporales.

A su vez, parte de los conocimientos requeridos en la fase de construcción no fueron satisfechos con la formación en la fase inicial por lo que se ha tenido que hacer pequeñas investigaciones durante la construcción para solventar esa carencia.

Tarea	Horas estimadas	Horas reales	Inicio estimado	Fin estimado	Inicio real	Fin real
TOTAL	235	520	28/2/2019	27/3/2019	14/2/2020	18/9/2020
Inicio de fase	0	0	28/2/2019	28/2/2019	14/2/2020	
Implementación del núcleo de la aplicación de acuerdo a los requisitos	100	220	28/2/2019	18/2/2019	14/2/2020	15/9/2020
Implementación básica de la interfaz gráfica	50	80	18/2/2019	6/3/2019	11/6/2020	15/9/2020
Implementación de casos de uso relacionados con preguntas	25	70	15/3/2019	22/3/2019	14/2/2020	28/8/2020
Implementación de casos de uso relacionados con escenarios	50	120	20/2/2019	15/3/2019	14/2/2020	18/9/2020
Implementación de casos de uso relacionados con configuraciones	10	30	22/3/2019	27/3/2019	13/9/2020	18/9/2020
Fin de fase	0	0	27/3/2019	27/3/2019	18/9/2020	18/9/2020

Tabla 3.7: Seguimiento de la fase de construcción

La fase de transición se ha ajustado a los tiempos previstos, aunque no arregla el retraso respecto a las fechas estimadas.

Tarea	Horas estimadas	Horas reales	Inicio estimado	Fin estimado	Inicio real	Fin real
TOTAL	10	6	27/3/2019	8/4/2019	18/9/2020	21/9/2020
Inicio de fase	0	0	27/3/2019	27/3/2019	18/9/2020	18/9/2020
Elaboración del manual de usuario	3	3	27/3/2019	8/4/2019	18/9/2020	21/9/2020
Elaboración del manual de instalación	2	1	27/3/2019	8/4/2019	18/9/2020	20/9/2020
Elaboración del manual de mantenimiento	5	2	27/3/2019	8/4/2019	18/9/2020	21/9/2020
Fin de fase	0	0	8/4/2019	8/4/2019	21/9/2020	21/9/2020

Tabla 3.8: Seguimiento de la fase de transición

3.4. Costes

Los costes totales del proyecto han sido de 0€. Todo el material usado por el alumno ya estaba amortizado y por tanto no se han generado costes adicionales. Se han usado también herramientas gratuitas para evitar costes adicionales.

Por el interés del análisis de costes, se procederá a simular los costes de este proyecto de haberse realizado fuera de un contexto académico.

Se supondrá un salario de programador Junior de 18.751€ anuales. Esto sería el un sueldo aproximado de unos 9,62€ la hora. Suponiendo un contrato flexible por horas hasta fin de obra, el precio del proyecto sería de 7.138€ de inversión en recursos humanos.

Capítulo 4

Análisis

4.1. Usuarios objetivo

La aplicación está enfocada al ámbito de la enseñanza y el aprendizaje. Es una herramienta de apoyo para complementar una formación sobre el uso del lenguaje SQL para obtener información de una base de datos.

Los usuarios que se tendrán en cuenta para la realización de la aplicación serán docentes o estudiantes. Los docentes usarán sobre todo la parte de la aplicación que permita crear entornos de prueba para los estudiantes. Los alumnos, a su vez, usarán la parte de la aplicación que permite contestar a un conjunto de preguntas preparadas previamente. Aún así, no se excluye la posibilidad de que estos usuarios usen la totalidad de la aplicación. En el caso de los profesores, podrán usar la parte de la aplicación centrada en la contestación de preguntas para probar los entornos creados. En el caso de los estudiantes podría ser interesante también que usaran la parte de gestión de escenarios para realizar autoevaluaciones o experimentar con sus propias bases de datos.

La terminología propia al ámbito de las bases de datos se supondrá conocida por el usuario. En el caso de los docentes, por ser el conocimiento que están impartiendo, y en el caso de los alumnos por formar parte de su formación. Aún así, se evitará cargar la aplicación con una excesiva carga teórica, puesto que el objetivo es apoyar mediante una experiencia práctica interactiva.

4.2. Diagrama de casos de uso

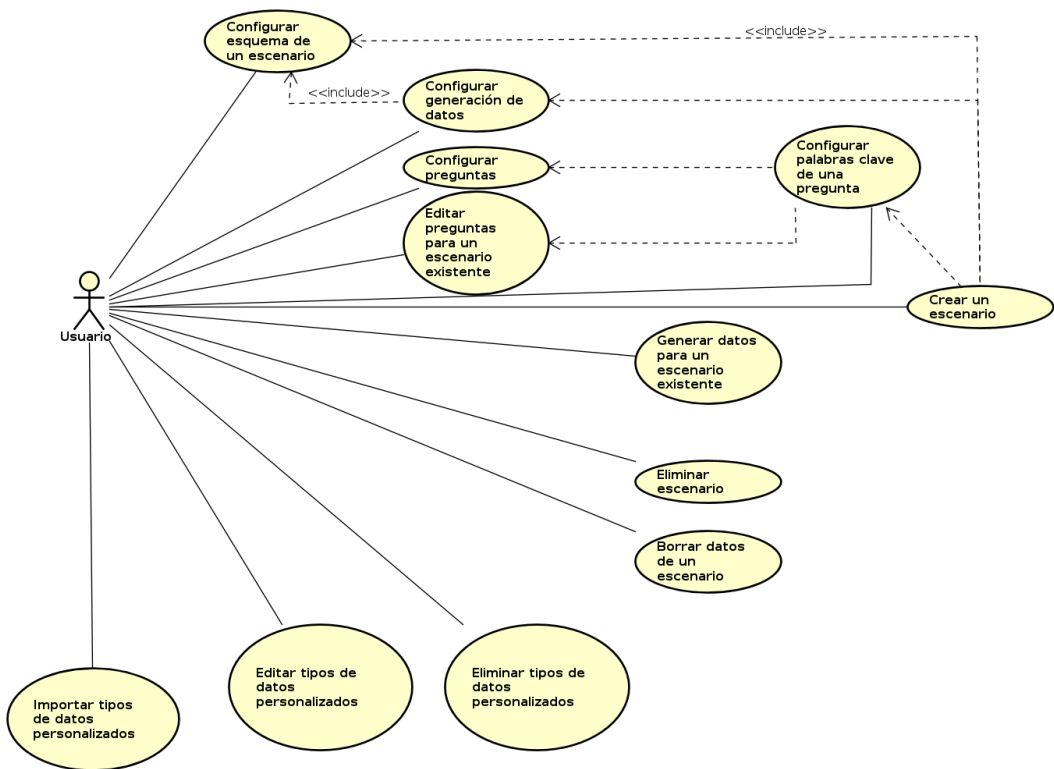


Figura 4.1: Diagrama de casos de uso relacionados con escenarios

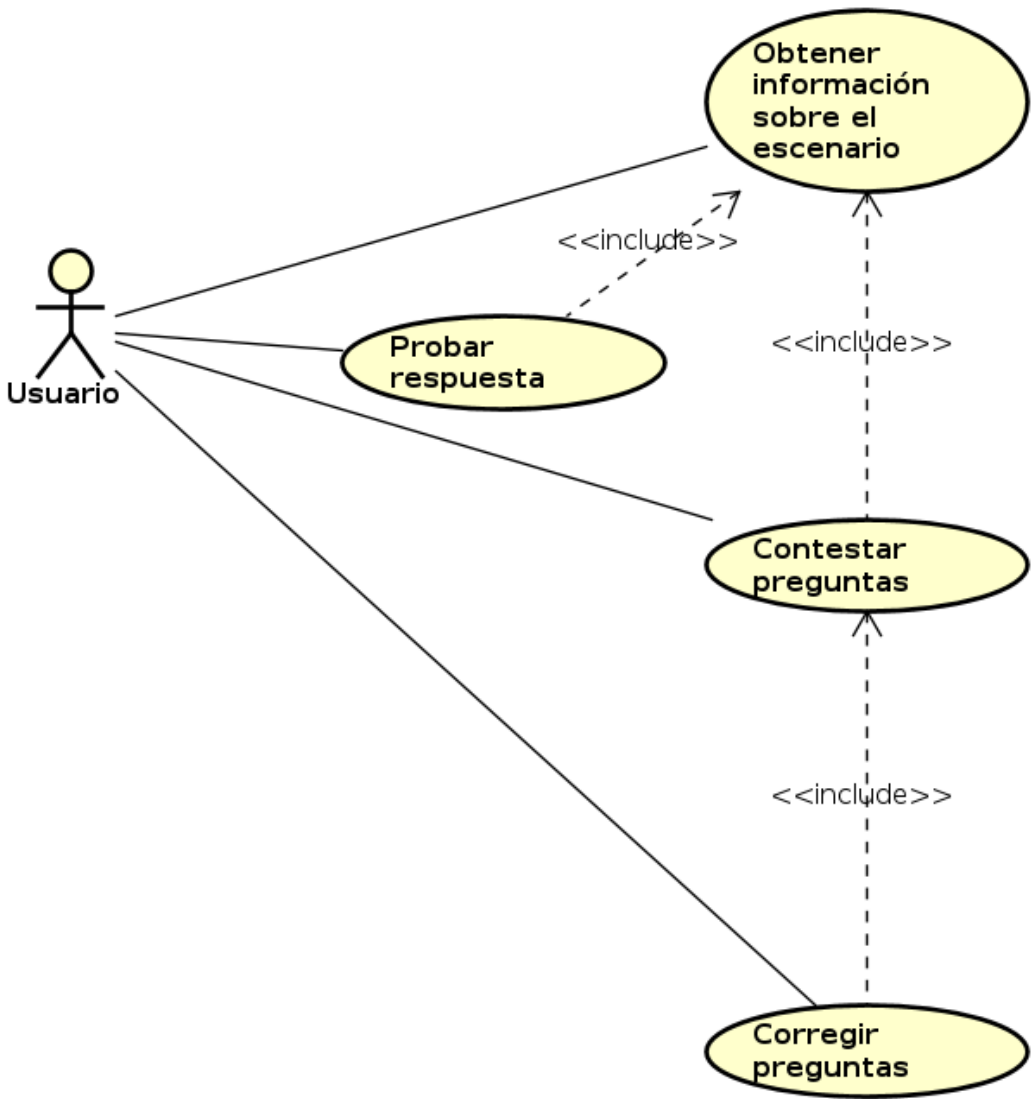


Figura 4.2: Diagrama de casos de uso relacionados con respuestas

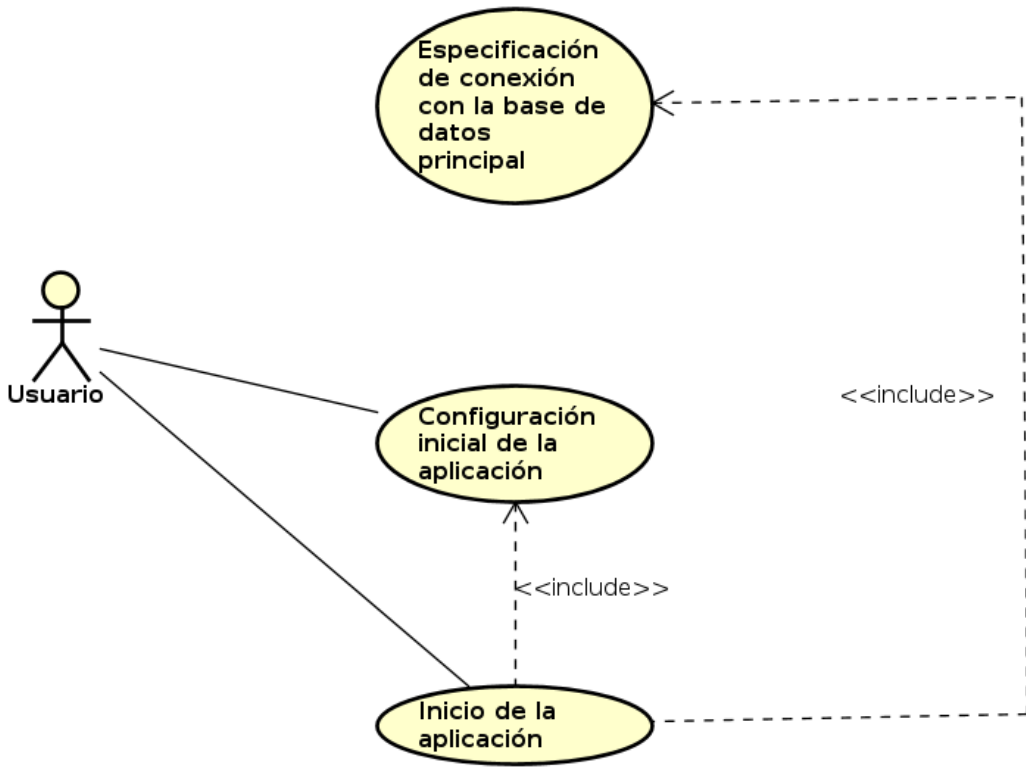


Figura 4.3: Diagrama de casos de uso relacionados con configuraciones

4.3. Modelo de dominio

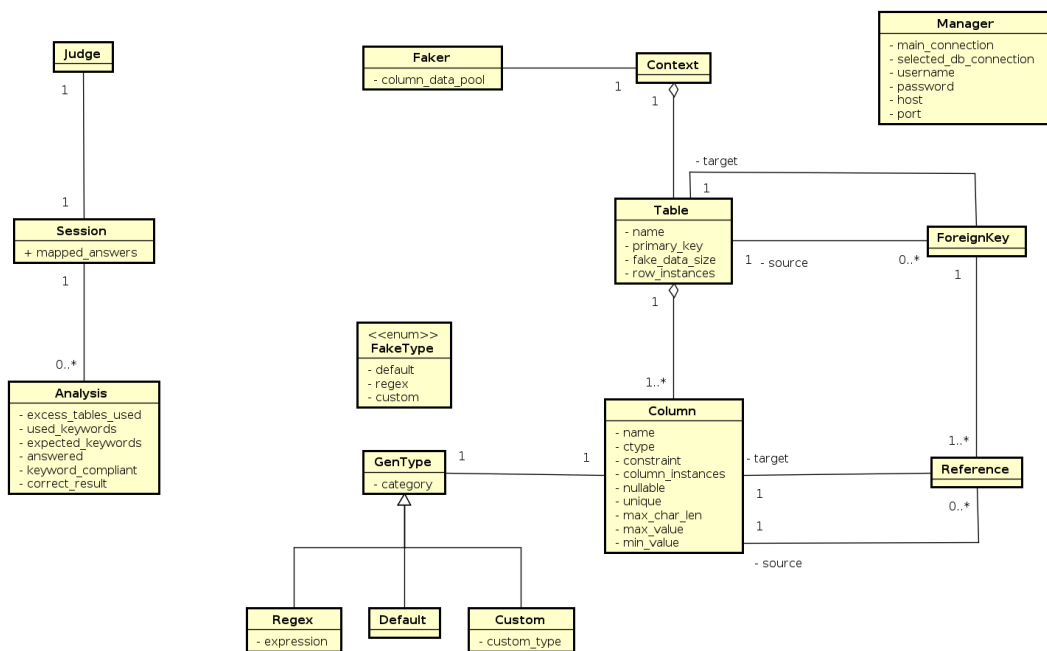


Figura 4.4: Modelo de dominio

Capítulo 5

Diseño

5.1. Arquitectura

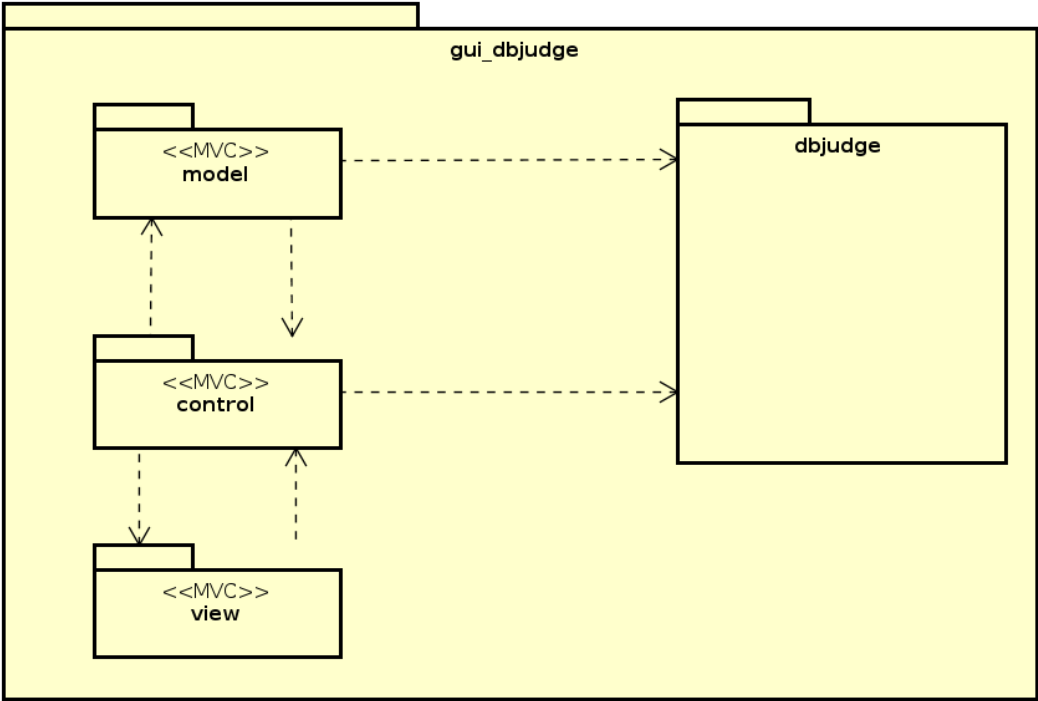


Figura 5.1: Arquitectura propuesta

5.2. Diseño detallado

A continuación se muestran los diagramas de clases detallados de aquellas clases que componen el núcleo de la aplicación.

La aplicación se ha desarrollado con Python mientras que el diseño se ha realizado con la herramienta Astah. Esta herramienta está centrada sobre todo en aplicaciones desarrolladas con c, c++ o Java. Algunos elementos pueden no corresponderse directamente con el comportamiento real de la aplicación.

Al ser un lenguaje con tipado dinámico, Python tiene algunas peculiaridades. Cabe destacar que no hay necesidad de usar *getters* o *setters* de forma directa, puesto que el estándar es utilizar los atributos directamente y en caso de necesitar cambiar como se modifican o como se obtienen, se utilizan unos elementos propios de Python llamados “descriptores”[8]. También hay que tener en cuenta que tanto funciones como atributos de clase son siempre accesibles aunque sean privados. Los mecanismos de Python respecto a esto se limitan solamente a evitar sobrescribir por error las propiedades privadas de una clase al heredar de ella.

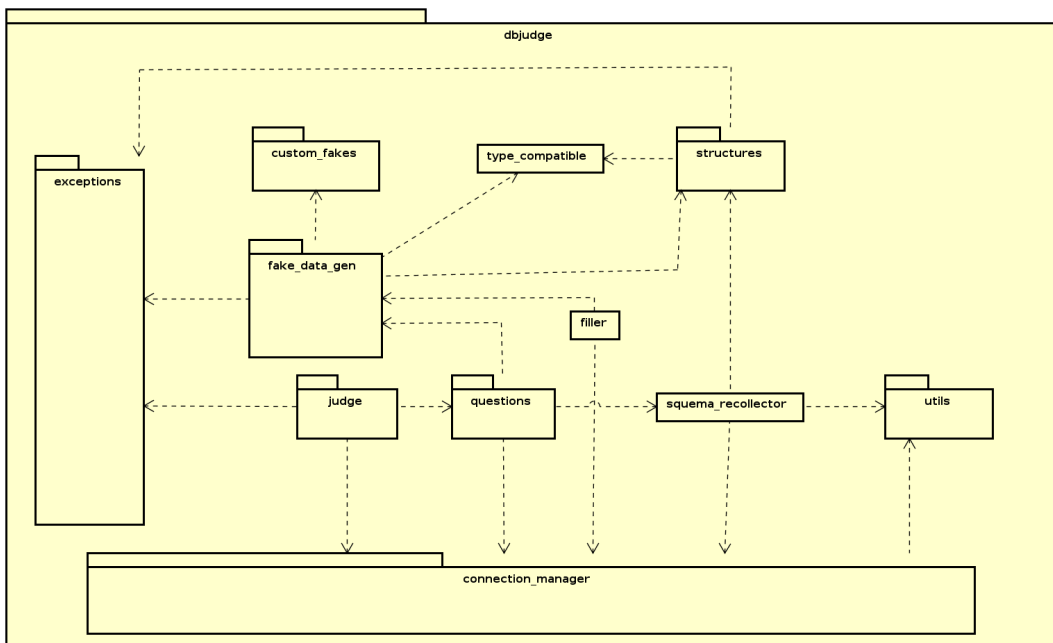


Figura 5.2: Diseño detallado de dbjudge

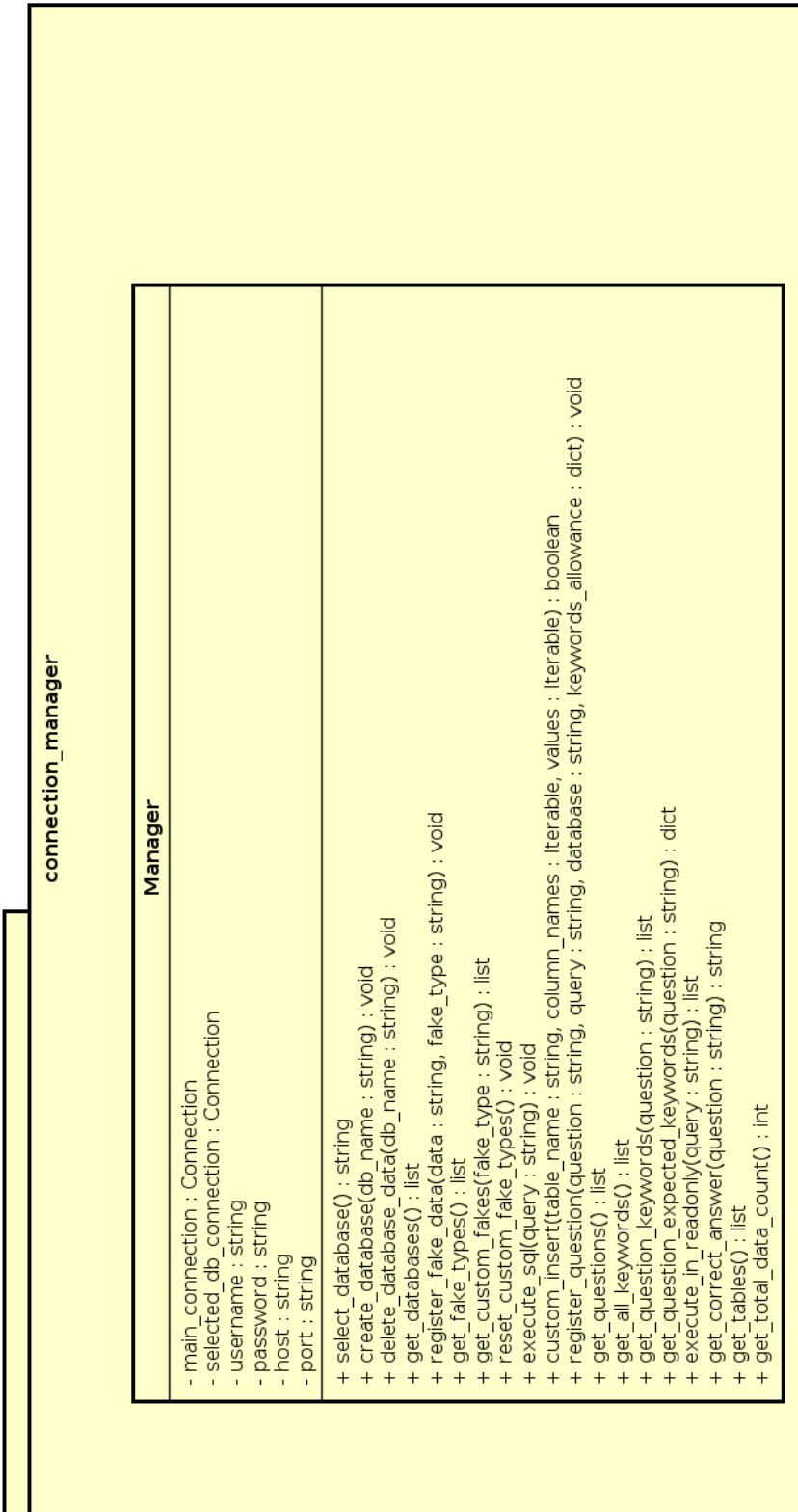


Figura 5.3: Diseño detallado de connection manager

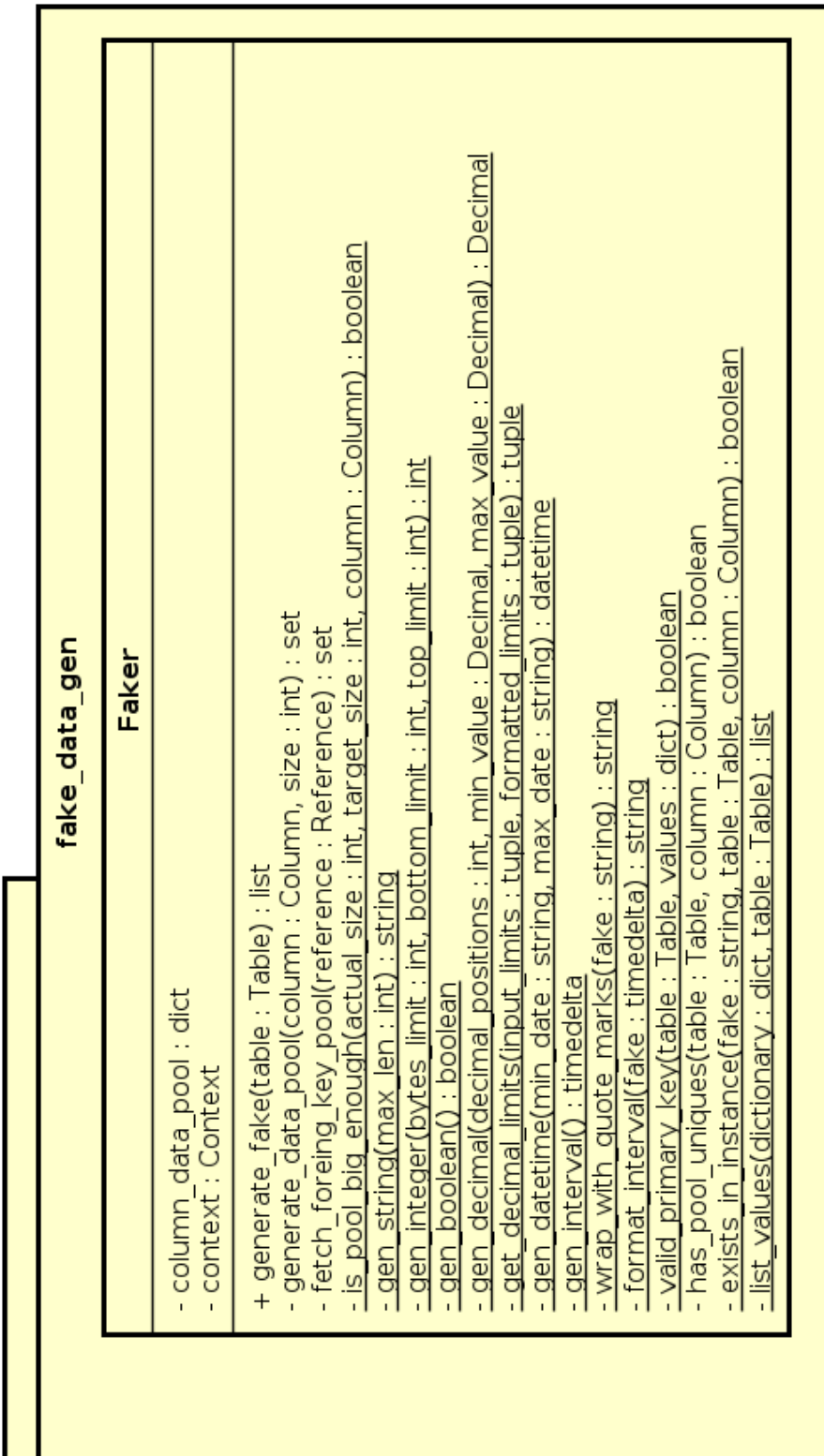


Figura 5.4: Diseño detallado de fake data gen

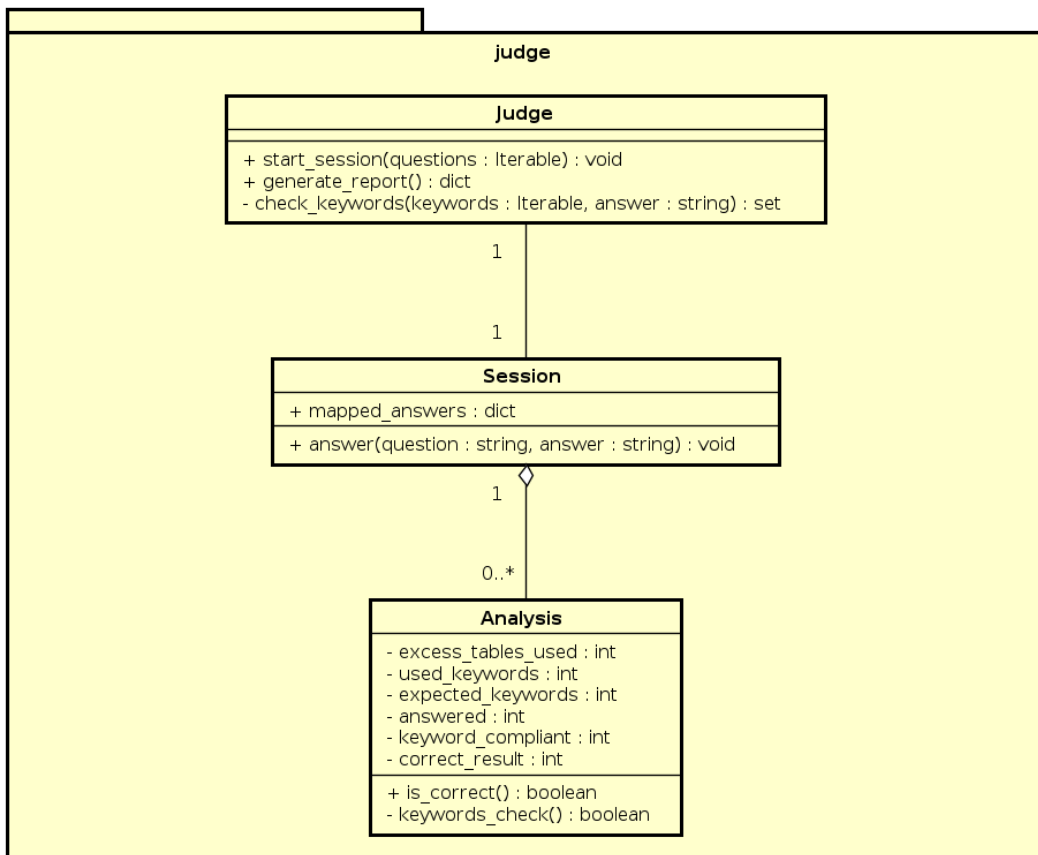


Figura 5.5: Diseño detallado de judge

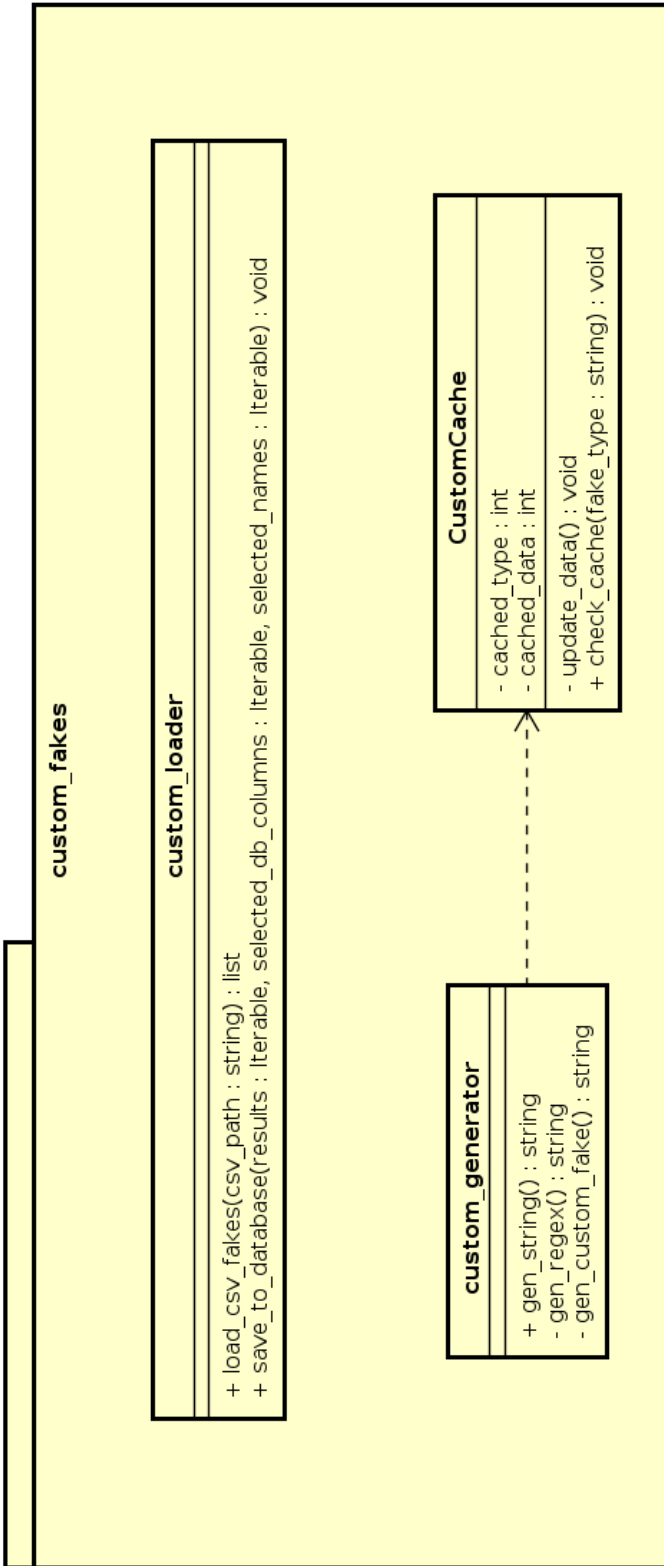


Figura 5.6: Diseño detallado de custom fakes

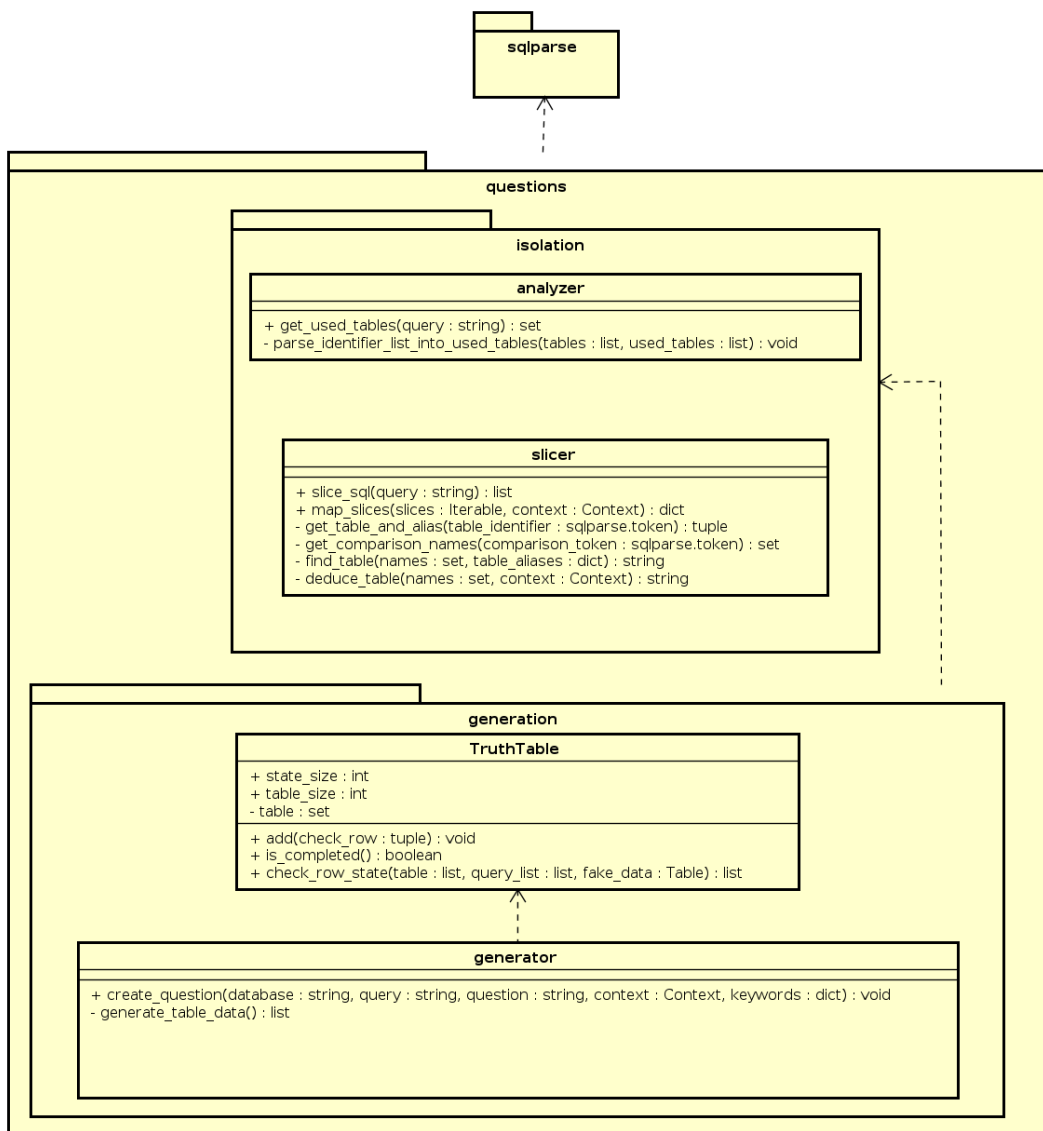


Figura 5.7: Diseño detallado de questions

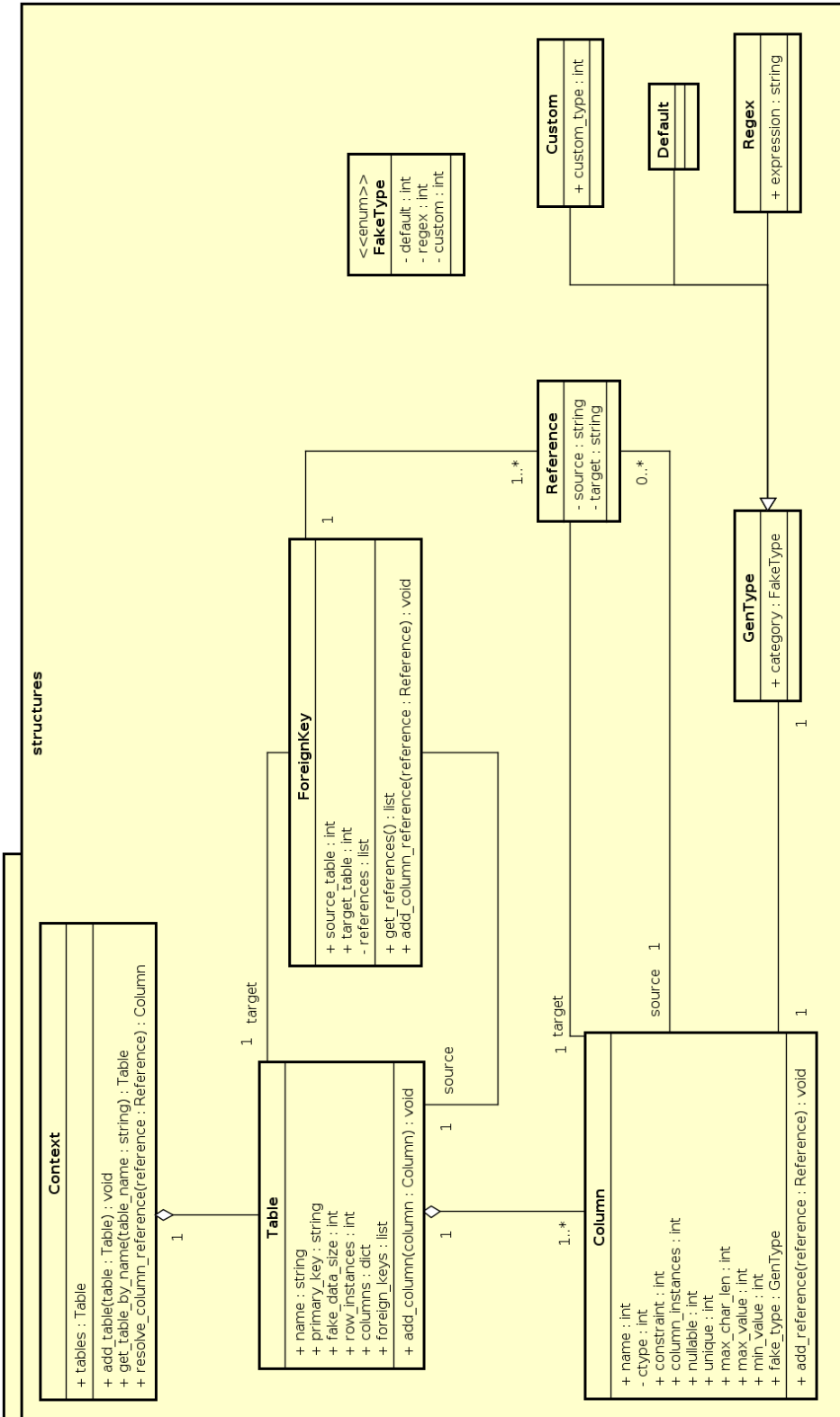


Figura 5.8: Diseño detallado de structures

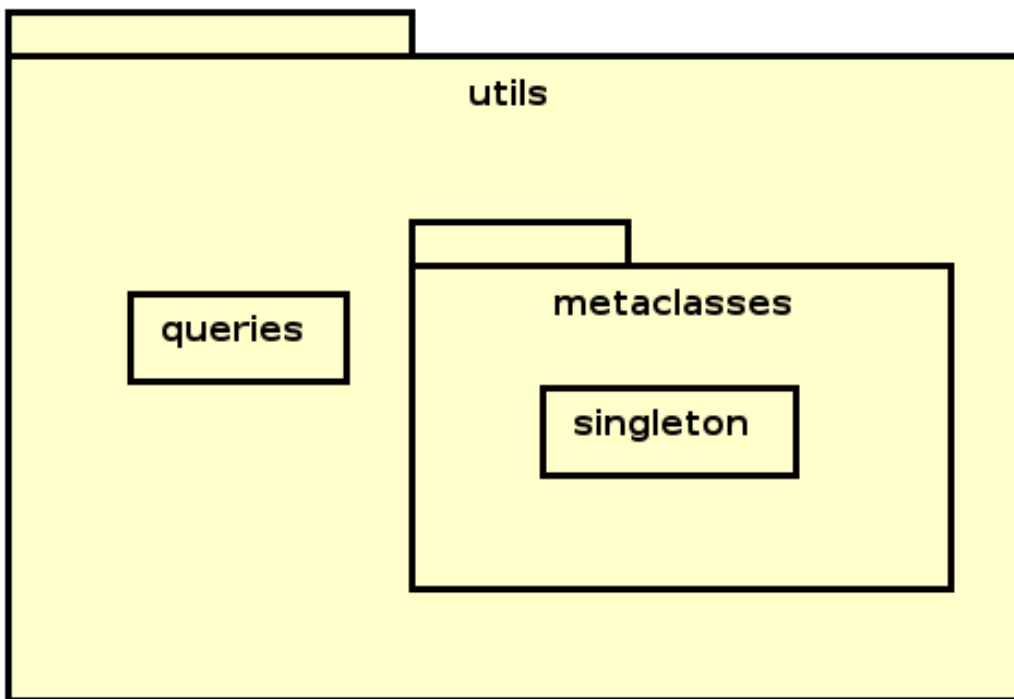


Figura 5.9: Diseño detallado de utils

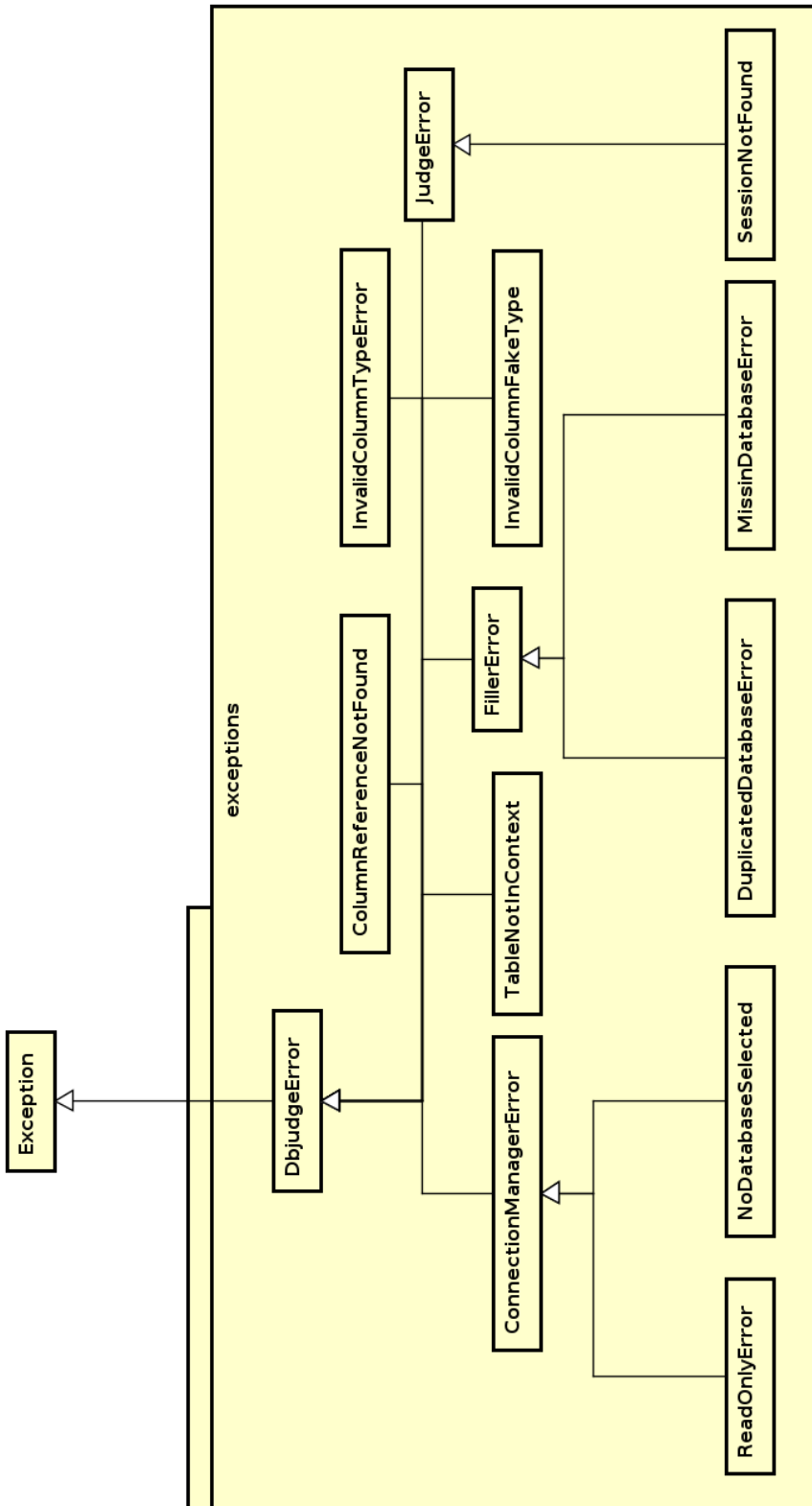


Figura 5.10: Diseño detallado de exceptions

5.3. Base de datos

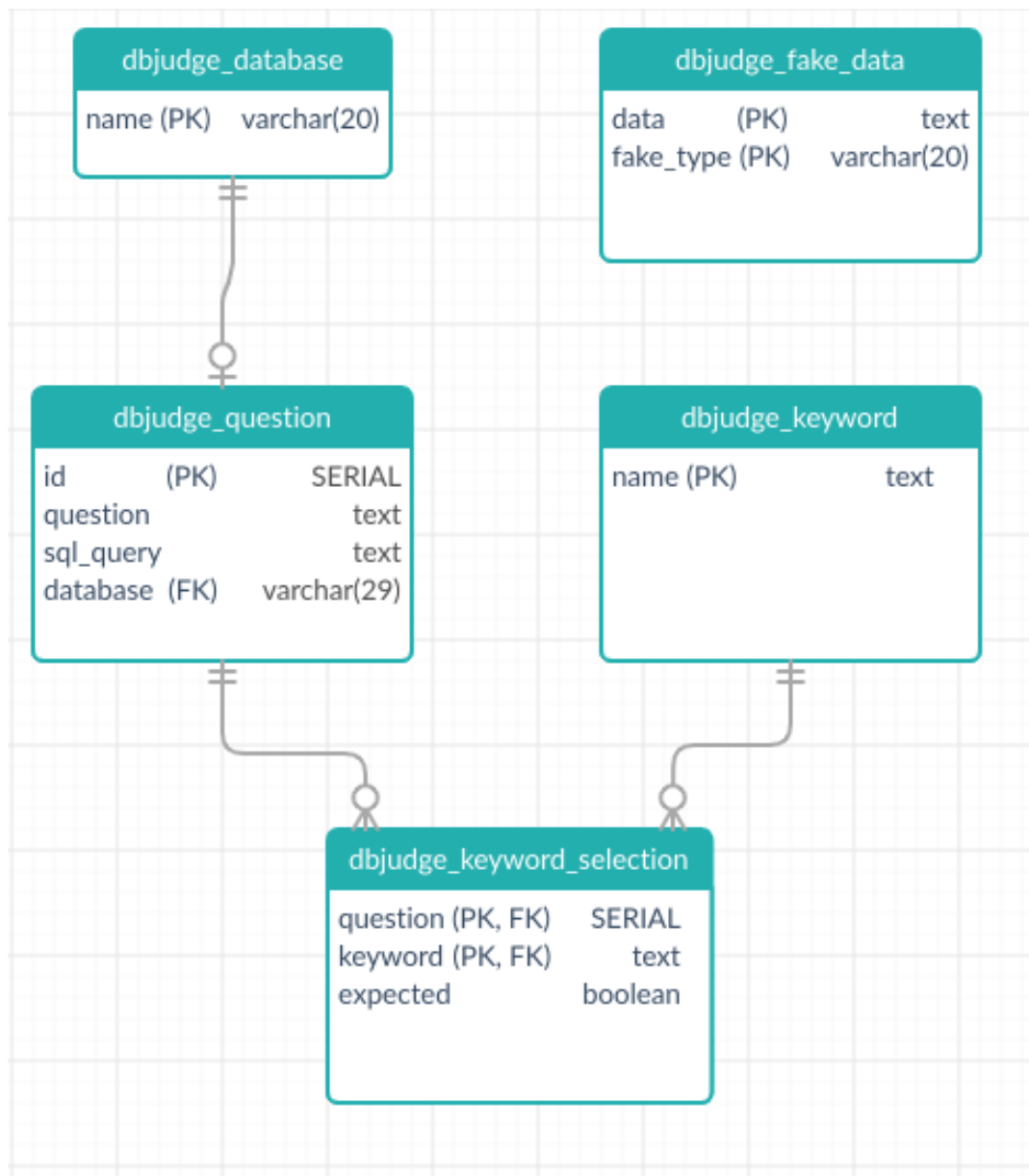


Figura 5.11: Modelo relacional

5.4. Interfaz gráfica

El proceso de diseño de la interfaz gráfica ha seguido un desarrollo iterativo. Se han ido realizando ligeras modificaciones según se han ido concretando las necesidades del cliente. Los bocetos se han realizado usando la herramienta *Balsamiq*[9]. Utilizar una herramienta digital de bocetaje como esta ha permitido realizar iteraciones más rápidas durante el proceso de diseño.

Se expondrán a continuación cada una de las vistas de acuerdo a la última iteración del bocetaje. La primera figura muestra la jerarquía de navegación dentro de la aplicación. Más adelante se detalla el aspecto general de las vistas que componen la interfaz.

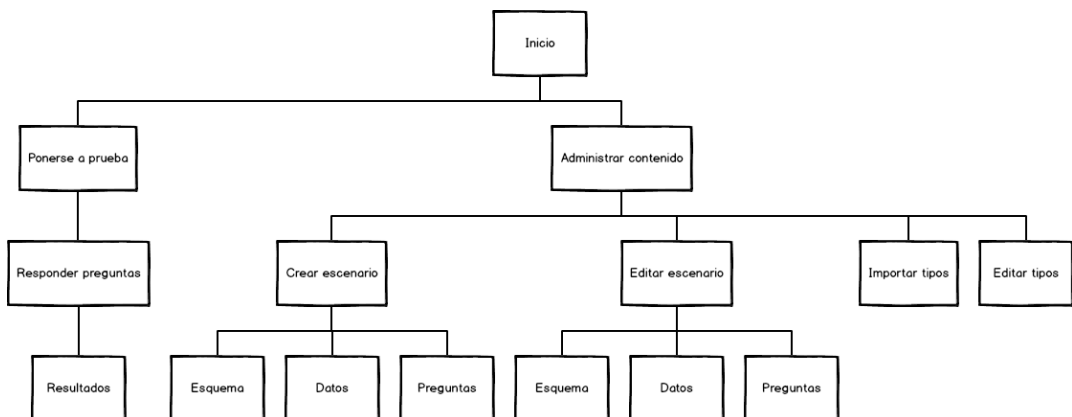


Figura 5.12: Estructura de vistas de la interfaz de usuario

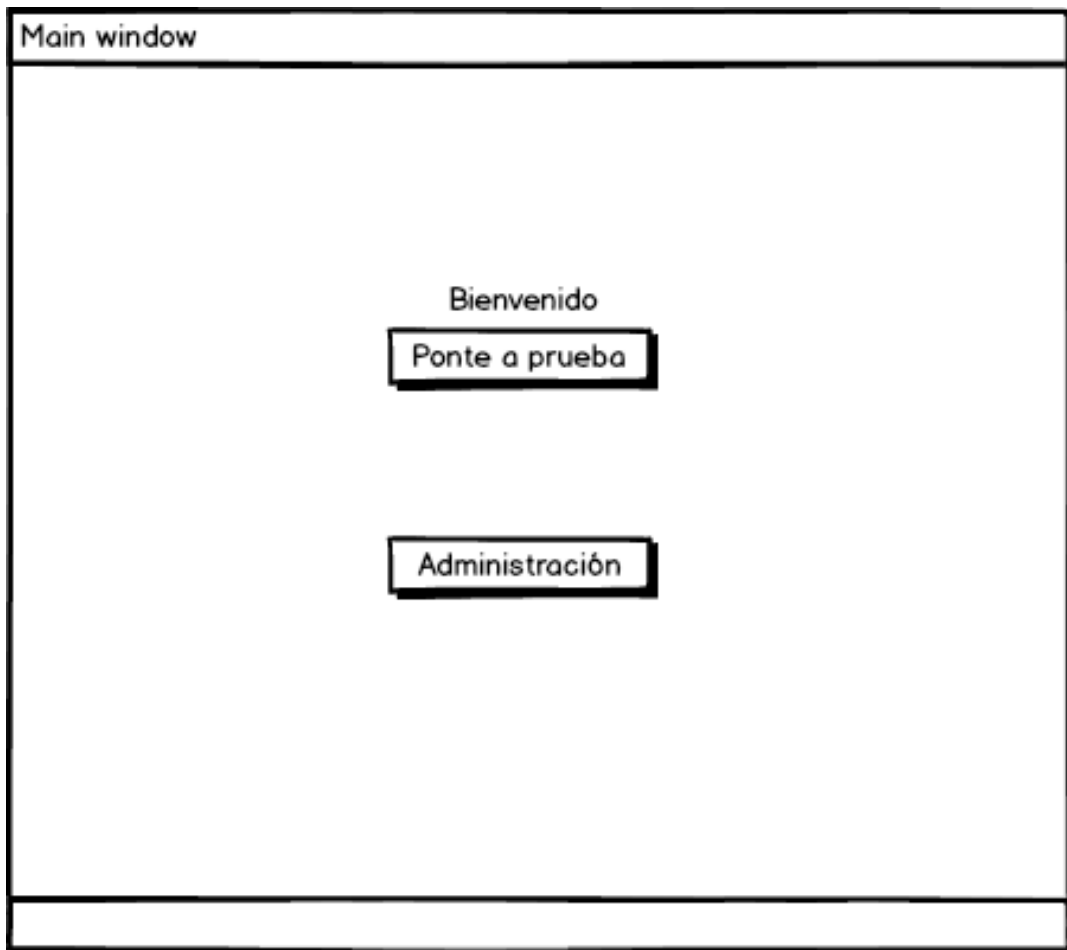


Figura 5.13: Menú principal

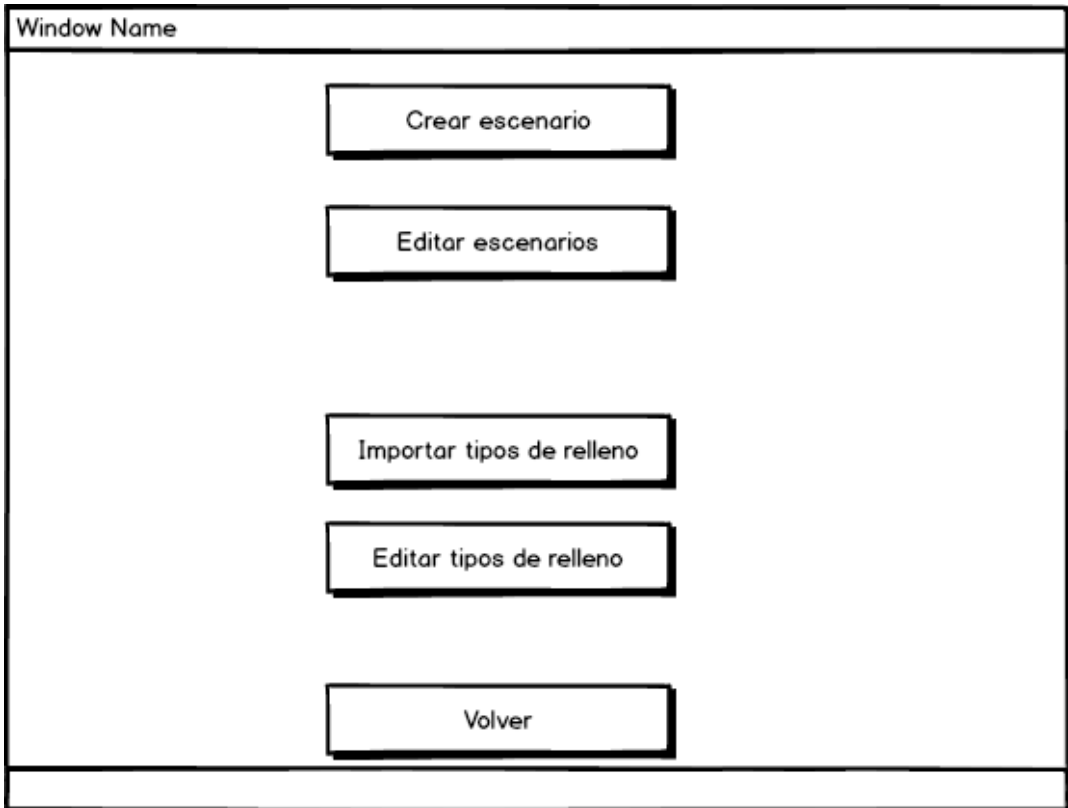



Figura 5.14: Menú de administración

Window Name

→ →

 Paso 1 - Elige el nombre del escenario e introduce el DDL

Numero de escenario:

Cargar definicion de tablas desde archivo

Ningun archivo seleccionado

Puedes introducir o modificar la definición manualmente en este área de texto:

Figura 5.15: Vista de creación de escenario - Paso 1

Window Name

Esquema → Generar datos → Configurar preguntas


 Paso 2 - Configura los datos para rellenar cada columna

Tabla 1 | Tabla 2 | Tabla 3 | Tabla 4

Cantidad de datos a generar:

Columna	Categoría	Tipo de dato	Longitud máxima:
Columna 1	<input type="radio"/> Regex	<input type="text" value="[a-zA-Z]+[1-9]*"/>	<input type="text"/>
	<input checked="" type="radio"/> Aleatorio	<input type="text" value="Valor máx"/> <input type="text" value="Valor min"/>	<input type="text"/>
	<input type="radio"/> Tipo personalizado	<input type="text" value="Tipo personalizado 1"/> ▼	
Columna 1	<input type="radio"/> Regex	<input type="text" value="[a-zA-Z]+[1-9]*"/>	<input type="text"/>
	<input checked="" type="radio"/> Aleatorio	<input type="text" value="Valor máx"/> <input type="text" value="Valor min"/>	<input type="text"/>
	<input type="radio"/> Tipo personalizado	<input type="text" value="Tipo personalizado 1"/> ▼	

Figura 5.16: Vista de creación de escenario - Paso 2

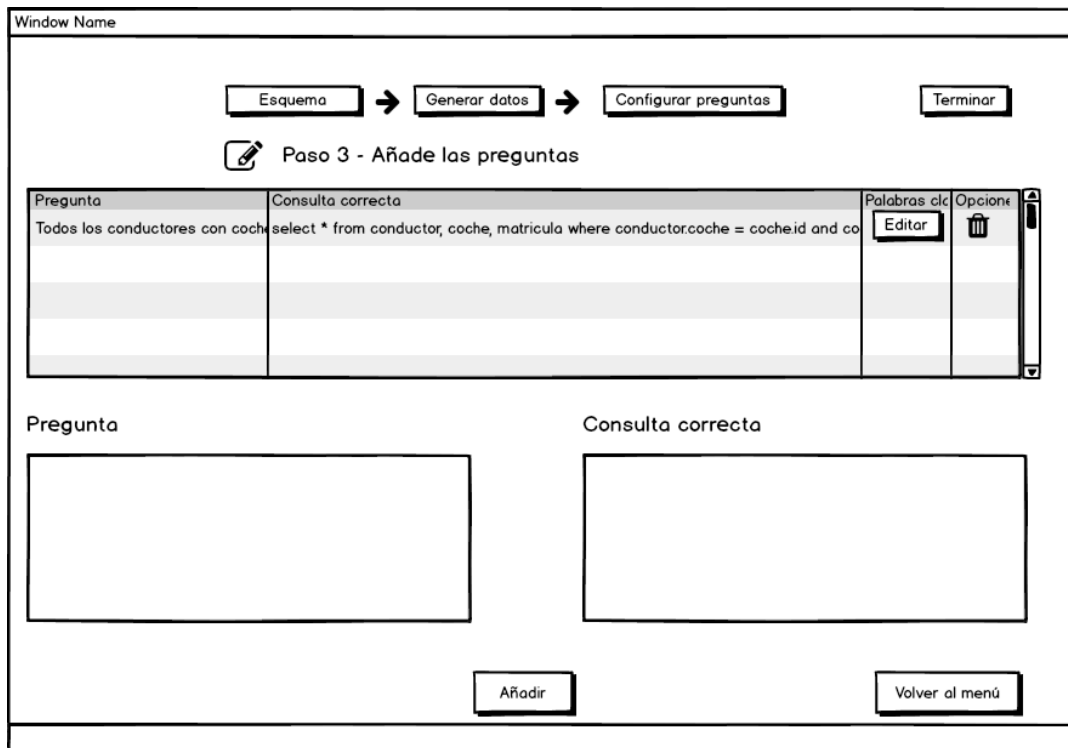


Figura 5.17: Vista de creación de escenario - Paso 3

Window Name

Esquema

Generar datos

Configurar preguntas

Escenario

Borrar datos generados

Eliminar escenario

Volver al menú

Figura 5.18: Vista de edición de escenario - General

Window Name

Esquema → Generar datos → Configurar preguntas

Seleccione escenario: Escenario [v] Generar datos

Tabla 1 | Tabla 2 | Tabla 3 | Tabla 4

Cantidad de datos a generar: 38

Columna	Categoría	Tipo de dato
Col1	<input type="radio"/> Regex	[a-zA-Z]+[1-9]* Longitud máxima: []
	<input checked="" type="radio"/> Aleatorio	10 [] 0 [] Longitud máxima: 13
	<input type="radio"/> Tipo personalizado	Tipo personalizado 1 [v]
Col2	<input type="radio"/> Regex	[a-zA-Z]+[1-9]* Longitud máxima: []
	<input checked="" type="radio"/> Aleatorio	Valor máx [] Valor mín [] Longitud máxima: []
	<input type="radio"/> Tipo personalizado	Tipo personalizado 1 [v]

Eliminar selección [] Volver al menú []

Figura 5.19: Vista de edición de escenario - Datos

Window Name

Esquema Generar datos Configurar preguntas

Seleccione escenario: Escenario Guardar preguntas

Pregunta	Consulta correcta	Palabras clave	Opciones
Todos los conductores con coche	select * from conductor, coche, matricula where conductor.coche = coche.id and co	Editar	Eliminar

Pregunta Consulta correcta

Añadir Volver al menú

Figura 5.20: Vista de edición de escenario - Preguntas

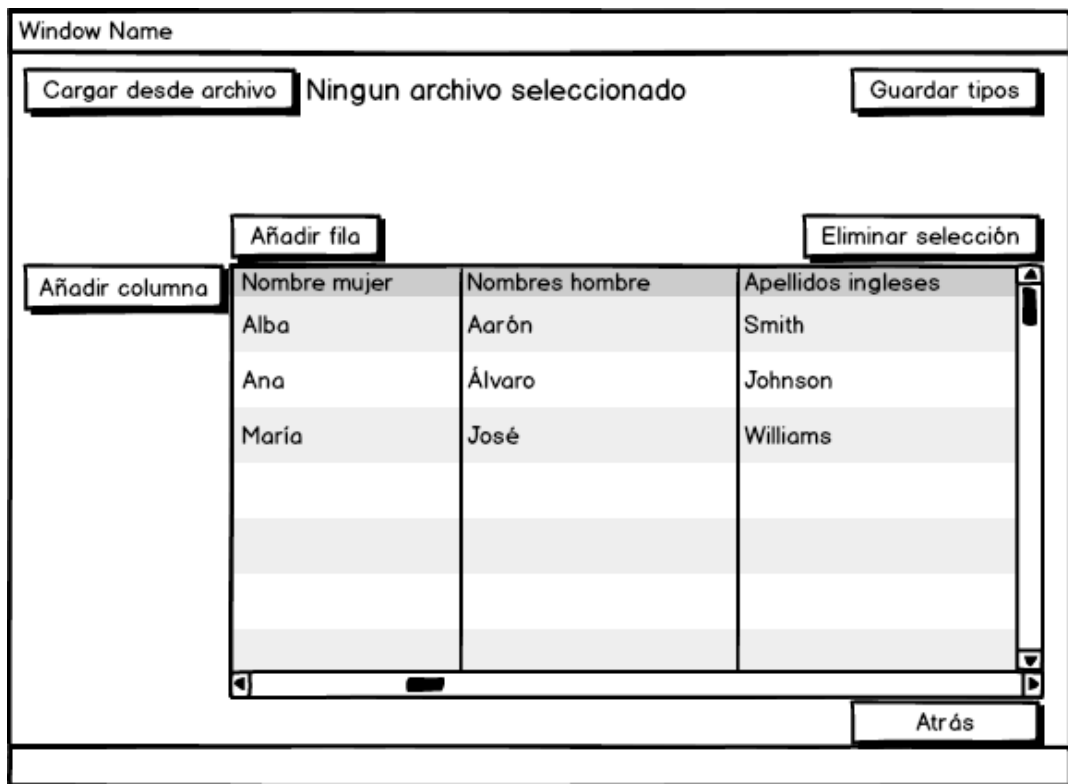


Figura 5.21: Vista para importar tipos de datos

Window Name

Guardar tipos

Eliminar selección

Nombre mujer	Nombres hombre	Apellidos ingleses
Alba	Aarón	Smith
Ana	Álvaro	Johnson
María	José	Williams

Atrás

Figura 5.22: Vista para editar tipos importados

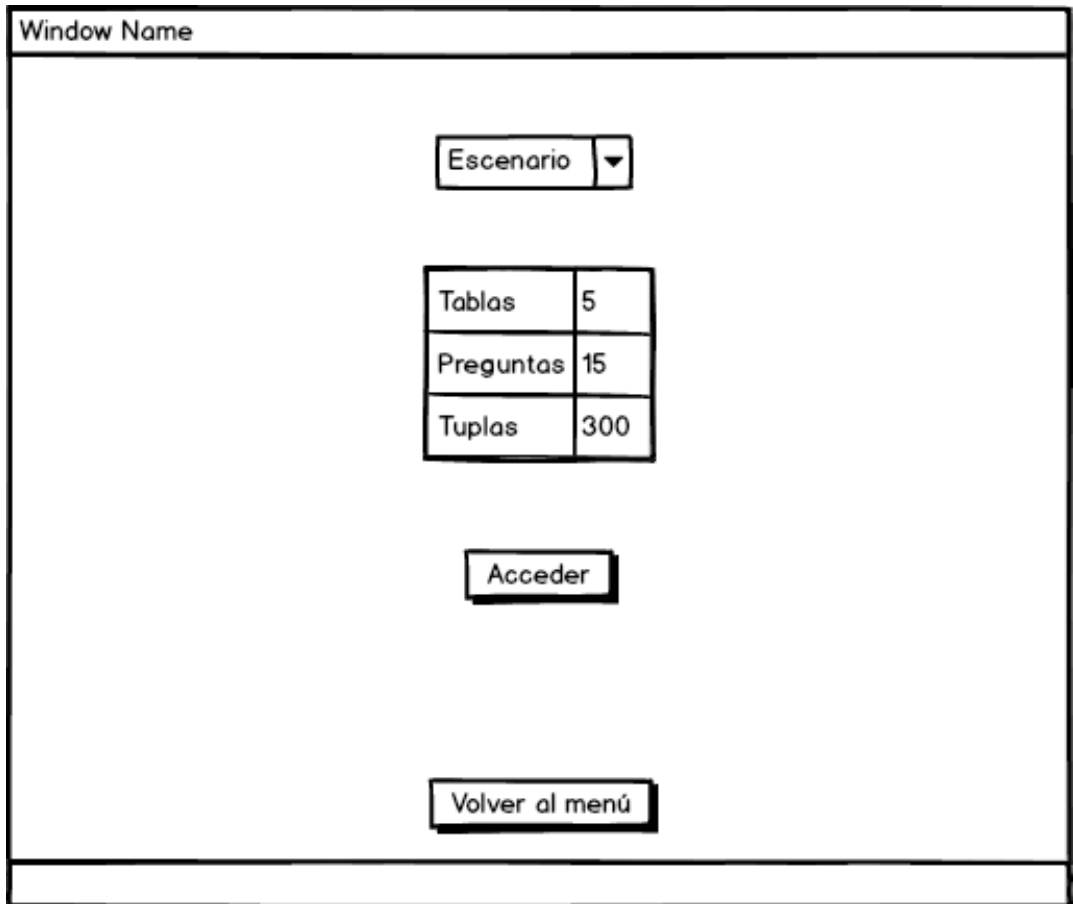


Figura 5.23: Menú de selección de pruebas

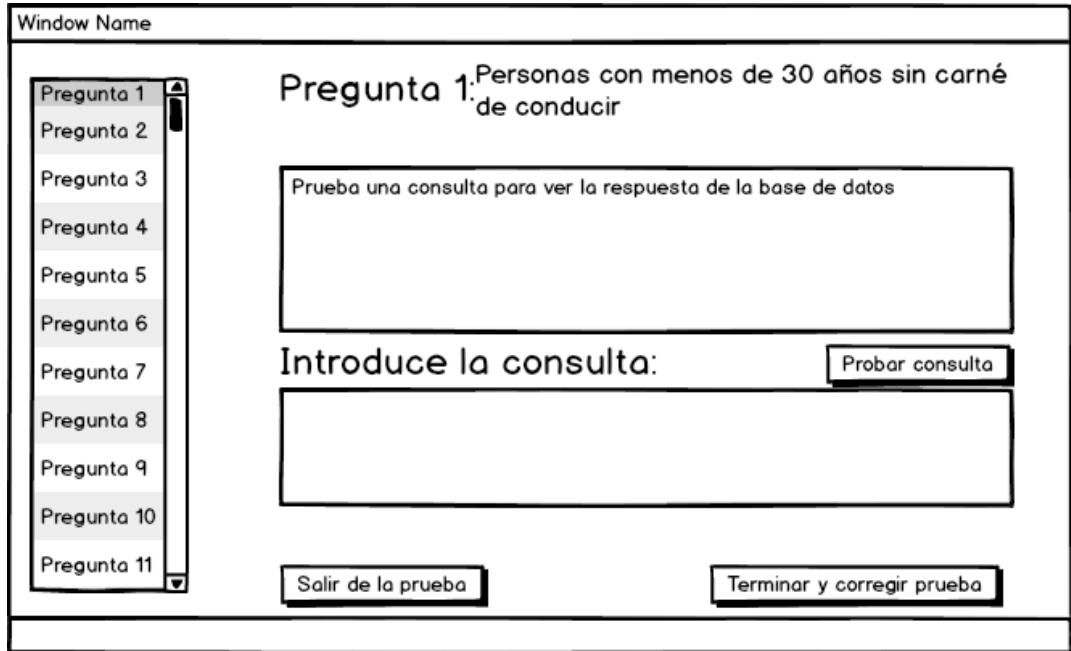


Figura 5.24: Vista de resolución de preguntas

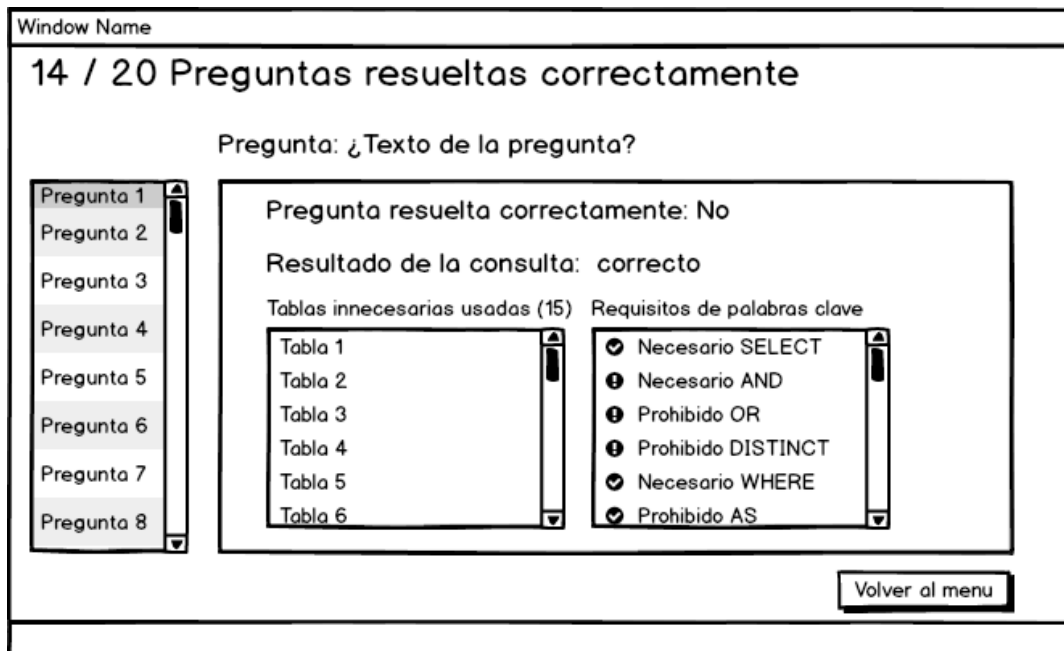


Figura 5.25: Vista de resultados de una prueba

Capítulo 6

Plan de pruebas

6.1. Diseño dirigido por pruebas

Exceptuando el desarrollo de la prueba de concepto al principio del proyecto, el desarrollo ha seguido un diseño dirigido por pruebas. No se ha aplicado de manera estricta, adaptándose a las necesidades del proyecto.

Para cada pequeña parte funcional, se realizaba una o varias pruebas previas que definiera el comportamiento deseado. Después se procedía a implementar la funcionalidad y se pasaban las pruebas para comprobar que no se hubieran cometido errores. Dichas pruebas han quedado almacenadas en una carpeta de tests. Estos tests permiten detectar cambios colaterales en la funcionalidad. También documentan de forma directa el uso esperado de la aplicación.

Este proceso ha sido exclusivo del desarrollo del núcleo de la aplicación. Durante el desarrollo de la interfaz gráfica se han usado pruebas manuales sobre los casos de uso. Se descartaron las pruebas automáticas de interfaz debido al retraso en los tiempos respecto a la planificación inicial del proyecto.

6.2. Pruebas unitarias

En esta sección se recogen enumeradas las pruebas automáticas que se han realizado durante el desarrollo. Las pruebas se han ejecutando con el framework *pytest*.

En cada una de las sub secciones se expondrán cada una de las pruebas con una descripción breve de la funcionalidad a probar y su resultado esperado. Las pruebas están agrupadas en archivos con terminación *_test.py*. Cada uno de estos archivos recoge las pruebas para uno o varios módulos del núcleo de la aplicación.

6.2.1. conftest

En el archivo *conftest.py* se recogen los métodos de preparación y limpieza del entorno de *testing*. Estos procesos de preparación y limpieza se ejecutan, dependiendo del método, una sola vez, una vez por módulo o una vez por cada prueba individual.

Identificador	Alcance	Descripción
PCF-1	Global	Prepara una conexión con la base de datos sin usar el <i>Manager</i> .

Identificador	Alcance	Descripción
PCF-2	Global	Prepara una instancia del <i>Manager</i> . Presupone un funcionamiento correcto de este.

Identificador	Alcance	Descripción
PCF-3	Global	Prepara un cursor de lectura-escritura a partir de la conexión creada por CNF-1.

Identificador	Alcance	Descripción
PCF-4	Global	Carga en la aplicación algunos tipos personalizados. Presupone un funcionamiento correcto del módulo <i>custom_fakes</i> .

Identificador	Alcance	Descripción
PCF-5	Módulo	Borra todos los elementos en la base de datos. Utiliza la conexión preparada en CNF-1.

Identificador	Alcance	Descripción
PCF-6	Prueba	Restaura los tipos personalizados que hayan podido ser eliminados después de una prueba.

Identificador	Alcance	Descripción
PCF-7	Prueba	Reinicia el <i>CustomCache</i> .

Identificador	Alcance	Descripción
PCF-8	Prueba	Crea una base de datos con un nombre específico dentro de la aplicación. Se asegura de eliminarla al finalizar la ejecución de la prueba Depende del <i>Manager</i> proporcionado por CNF-2.

Identificador	Alcance	Descripción
PCF-9	Prueba	Reinicia completamente la base de datos principal. Depende del <i>Manager</i> proporcionado por CNF-2.

Identificador	Alcance	Descripción
PCF-10	Prueba	Crea una tabla vacía con un nombre específico en la base de datos principal. Depende del <i>Manager</i> proporcionado por CNF-2.

6.2.2. column_test

Identificador	PU-1
Descripción	Intenta crear una columna con un tipo no soportado.
Resultado esperado	Se produce una excepción <i>InvalidColumnTypeError</i> .

6.2.3. analyzer_test

Identificador	PU-2
Descripción	Obtener las tablas usadas por la siguiente consulta: Select lib.titulo from Libro lib, Prestamo, Persona Where lib.ejemplares > 10;
Resultado esperado	Libro, Prestamo, Persona

Identificador	PU-3
Descripción	Obtener las tablas usadas por la siguiente consulta: select * from Libro INNER JOIN Prestamo ON Libro.titulo=Prestamo.titulo;
Resultado esperado	Libro, Prestamo

6.2.4. connection_manager_test

Identificador	PU-4
Descripción	Comprueba que la aplicación ha creado las tablas necesarias una vez iniciada.
Resultado esperado	La aplicación está instalada correctamente.

Identificador	PU-5
Descripción	Comprueba que la aplicación ha creado las tablas necesarias una vez iniciada la aplicación desde otra instancia.
Resultado esperado	La aplicación está instalada correctamente.

Identificador	PU-6
Descripción	Instancia el <i>Manager</i> 2 veces.
Resultado esperado	Se referencia a un único <i>Manager</i> .

Identificador	PU-7
Descripción	Instancia el <i>Manager</i> 2 veces.
Resultado esperado	Ninguna referencia desaparece o hace que el <i>Manager</i> cambie de estado.

Identificador	PU-8
Descripción	Intenta borrar el <i>Manager</i> .
Resultado esperado	La conexión del manager con la base de datos se cierra correctamente.

Identificador	PU-9
Descripción	Intenta borrar el <i>Manager</i> con una base de datos seleccionada.
Resultado esperado	La conexión principal del <i>Manager</i> con la base de datos se cierra correctamente. La conexión con la base de datos seleccionada se cierra correctamente.

Identificador	PU-10
Descripción	Intenta crear una nueva base de datos.
Resultado esperado	Una base de datos se nueva se ha creado y añadido a la tabla <i>dbjudge_database</i> .

Identificador	PU-11
Descripción	Intenta crear una base de datos con el nombre de una existente.
Resultado esperado	Se produce una excepción de tipo <i>DuplicatedDatabaseError</i> .

Identificador	PU-12
Descripción	Intenta borrar una base de datos.
Resultado esperado	La base de datos es borrada correctamente. Se elimina de la tabla <i>dbjudge_database</i> correctamente.

Identificador	PU-13
Descripción	Intenta borrar una base de datos que no existe.
Resultado esperado	Se produce una excepción de tipo <i>MissingDatabaseError</i> .

Identificador	PU-14
Descripción	Crea 3 bases de datos y recupera la lista de bases de datos disponibles.
Resultado esperado	Aparecen en la lista únicamente las 3 bases de datos creadas.

Identificador	PU-15
Descripción	Selecciona una base de datos concreta.
Resultado esperado	La conexión de selección apunta a la base de datos correcta.

Identificador	PU-16
Descripción	Selecciona una base de datos e intenta ejecutar una consulta.
Resultado esperado	La consulta se realiza correctamente en la base de datos seleccionada.

Identificador	PU-17
Descripción	Intenta seleccionar una base de datos inexistente.
Resultado esperado	Se produce una excepción de tipo <i>MissingDatabaseError</i> .

Identificador	PU-18
Descripción	Se recupera la lista de tipos personalizados.
Resultado esperado	Aparecen únicamente los tipos cargados en PCF-4.

Identificador	PU-19
Descripción	Se recupera la lista de datos de un tipo personalizado concreto.
Resultado esperado	Los datos coinciden con los datos cargados de ese tipo en PCF-4.

Identificador	PU-20
Descripción	Se intentan borrar los tipos personalizados.
Resultado esperado	Los tipos personalizados se borran correctamente.

Identificador	PU-21
Descripción	Se intenta hacer una inserción en una tabla sin necesidad de proveer un cursor.
Resultado esperado	El número de tuplas en la tabla coincide con la cantidad insertada.

Identificador	PU-22
Descripción	Intenta recuperar el enunciado de varias preguntas.
Resultado esperado	Los enunciados coinciden con los nombres esperados.

Identificador	PU-23
Descripción	Intenta registrar una pregunta de forma repetida.
Resultado esperado	La pregunta solo se registra una vez.

Identificador	PU-24
Descripción	Intenta obtener la lista de tablas de una base de datos.
Resultado esperado	Las nombres de las tablas coinciden con los esperados.

Identificador	PU-25
Descripción	Intenta obtener el número total de tuplas de una base de datos.
Resultado esperado	Obtiene la cantidad de tuplas almacenadas en esa base de datos.

Identificador	PU-26
Descripción	Intenta ejecutar una consulta no soportada en el modo solo lectura.
Resultado esperado	Se produce una excepción de tipo <i>ReadOnlyError</i> .

Identificador	PU-27
Descripción	Intenta obtener la lista de palabras clave.
Resultado esperado	Las palabras clave son las esperadas.

Identificador	PU-28
Descripción	Intenta eliminar todas las tuplas de una base de datos.
Resultado esperado	La base de datos objetivo queda vacía.

Identificador	PU-29
Descripción	Intenta obtener la respuesta correcta a una pregunta.
Resultado esperado	La respuesta correcta coincide con la esperada.

6.2.5. context_test

Identificador	PU-30
Descripción	Se intenta obtener una tabla que no está presente en el contexto actual.
Resultado esperado	Se produce una excepción de tipo <i>TableNotInContext</i> .

Identificador	PU-31
Descripción	Se intenta resolver una referencia que no está en el contexto.
Resultado esperado	Se produce una excepción de tipo <i>ColumnReferenceNotFound</i> .

6.2.6. custom_fakes_test

Identificador	PU-32
Descripción	Se intenta establecer una configuración de generación inválida.
Resultado esperado	Se produce una excepción de tipo <i>InvalidColumnFakeType</i> .

Identificador	PU-33
Descripción	Se intenta generar una cadena de caracteres siguiendo la expresión regular <code>[ab]+</code> con un límite de caracteres.
Resultado esperado	Una cadena de caracteres formada por varios caracteres a y b con una longitud menor al límite impuesto.

Identificador	PU-34
Descripción	Se reinicia la instancia de <i>CustomCache</i> . Se establece un nuevo tipo de datos para almacenar.
Resultado esperado	Los datos almacenados antes del reinicio no están presentes en el caché. Los nuevos datos están presentes en caché.

Identificador	PU-35
Descripción	Se generan datos personalizados hasta agotar la cantidad de datos únicos.
Resultado esperado	Los datos obtenidos coinciden con los datos precargados.

Identificador	PU-36
Descripción	Se intentan recuperar los datos de uno de los tipos personalizados.
Resultado esperado	Los datos se recuperan correctamente.

Identificador	PU-37
Descripción	Se intenta cargar un tipo personalizado especificando el nombre.
Resultado esperado	Los datos se guardan correctamente con el tipo especificado.

Identificador	PU-38
Descripción	Se intenta cargar un tipo personalizado sin especificar nada.
Resultado esperado	Los datos se guardan correctamente usando la primera fila del archivo CSV como nombre del tipo.

Identificador	PU-39
Descripción	Se intentan cargar varios tipos personalizados especificando qué columnas del archivo CSV se querían guardar.
Resultado esperado	Los datos se guardan correctamente. Los tipos no especificados no se guardan en la base de datos.

6.2.7. filler_test

Identificador	PU-40
Descripción	Se intenta rellenar una base de datos con datos aleatorios.
Resultado esperado	La operación transcurre sin producir excepciones no controladas.

Identificador	PU-41
Descripción	Se intenta rellenar una base de datos con una cantidad concreta de datos para cada tabla.
Resultado esperado	La cantidad de datos generados se corresponde con la especificada.

6.2.8. judge_test

Identificador	PU-42
Descripción	Se intenta generar un reporte de una sesión.
Resultado esperado	Los resultados en el reporte son los esperados.

Identificador	PU-43
Descripción	Se intenta generar un reporte con exceso de tablas.
Resultado esperado	El exceso de tablas se refleja correctamente en el reporte.

Identificador	PU-44
Descripción	Se intenta generar un reporte con repuestas que usen palabras clave.
Resultado esperado	El uso de las palabras clave se refleja correctamente en el reporte.

Identificador	PU-45
Descripción	Se intenta generar un reporte con preguntas sin contestar.
Resultado esperado	Se detecta correctamente que preguntas están sin contestar. El resultado de las preguntas sin contestar no es correcto.

Identificador	PU-46
Descripción	Se intenta generar un reporte sin que el juez haya iniciado una sesión.
Resultado esperado	Se produce una excepción de tipo <i>SessionNotFound</i> .

Identificador	PU-47
Descripción	Se intenta crear una nueva sesión.
Resultado esperado	La sesión anterior es eliminada correctamente. La nueva sesión se crea correctamente. La nueva sesión no tiene preguntas contestadas.

Identificador	PU-48
Descripción	Se intenta responder a una pregunta.
Resultado esperado	La respuesta se guarda correctamente asociada a su pregunta.

Identificador	PU-49
Descripción	Se intenta detectar una cadena de texto vacía como palabra clave.
Resultado esperado	No se detecta su uso.

Identificador	PU-50
Descripción	Se intenta generar un reporte con una pregunta que se ha respondido con una consulta incorrecta sintácticamente.
Resultado esperado	La pregunta se considera mal contestada y con un resultado incorrecto.

Identificador	PU-51
Descripción	Se intenta responder con una consulta no permitida.
Resultado esperado	El reporte muestra la pregunta como incorrecta y con un resultado incorrecto.

6.2.9. question_test

Identificador	PU-52
Descripción	Se intenta dividir la siguiente consulta en consultas más simples: SELECT * FROM person p WHERE p.age > 20 and p.name not like 'juan' and NOT p.color like 'red' and p.adress < 30000;
Resultado esperado	SELECT * FROM person p WHERE p.age > 20 ; - SELECT * FROM person p WHERE p.name not like 'juan' ; - SELECT * FROM person p WHERE NOT p.color like 'red' ; - SELECT * FROM person p WHERE p.adress < 30000;

Identificador	PU-53
Descripción	Se intenta realizar un mapeado de varias consultas simples.
Resultado esperado	Las consultas simples son asociadas a una de las tablas que referencian.

Identificador	PU-54
Descripción	Se crea y rellena una <i>TruthTable</i> .
Resultado esperado	Al completarse no admite más estados y cambia de estado a completa.

Identificador	PU-55
Descripción	Se intenta generar datos a partir de una consulta simple.
Resultado esperado	Se generan datos correctamente.

Identificador	PU-56
Descripción	Se intenta crear una pregunta.
Resultado esperado	La pregunta se crea correctamente. La respuesta guardada se corresponde con la especificada. Se han generado datos adicionales a partir de consultas simples.

Identificador	PU-57
Descripción	Se intenta encontrar una tabla no referenciada.
Resultado esperado	No se encuentra la tabla, el resultado es una lista vacía.

6.2.10. `squema_recolector_test`

Identificador	PU-58
Descripción	Se intenta obtener el contexto actual a partir de una base de datos.
Resultado esperado	El contexto contiene el número esperado de tablas, columnas y claves foráneas.

Identificador	PU-59
Descripción	Se intentan cargar en un <i>Context</i> las tablas de una base de datos.
Resultado esperado	Las tablas cargadas son las esperadas.

Identificador	PU-60
Descripción	Se intentan cargar en una <i>Table</i> las columnas de dicha tabla desde la base de datos.
Resultado esperado	Las columnas cargadas son las esperadas.

Identificador	PU-61
Descripción	Se intentan cargar las referencias dentro de un <i>Context</i> que ya tiene cargadas tablas y columnas.
Resultado esperado	Las referencias se cargan correctamente.

Identificador	PU-62
Descripción	Se intenta formatear la salida de una clave primaria de la base de datos.
Resultado esperado	El formateo se realiza correctamente.

Identificador	PU-63
Descripción	Se intenta obtener las columnas definidas como <i>UNIQUE</i> .
Resultado esperado	Se obtienen las columnas definidas como <i>UNIQUE</i> . Las claves primarias no aparecen.

Identificador	PU-64
Descripción	Se actualizan las instancias del contexto actual.
Resultado esperado	Las nuevas instancias de las columnas se actualizan correctamente.

6.2.11. `type_compatible_test`

Identificador	PU-65
Descripción	Se comprueban los tipos compatibles de Postgresql con cadenas de caracteres.
Resultado esperado	Son compatibles <i>character</i> , <i>character varying</i> y <i>national character</i> .

Identificador	PU-66
Descripción	Se comprueban los tipos compatibles de Postgresql con números enteros.
Resultado esperado	Son compatibles <i>integer</i> y <i>smallint</i> .

Identificador	PU-67
Descripción	Se comprueban los tipos compatibles de Postgresql con números decimales.
Resultado esperado	Es compatible <i>real</i> .

Identificador	PU-68
Descripción	Se comprueban los tipos compatibles de Postgresql con intervalos.
Resultado esperado	Es compatible <i>interval</i> .

Identificador	PU-69
Descripción	Se comprueban los tipos compatibles de Postgresql con valores booleanos.
Resultado esperado	Es compatible <i>boolean</i> .

Identificador	PU-70
Descripción	Se comprueban los tipos compatibles de Postgresql con fechas.
Resultado esperado	Son compatibles <i>date</i> , <i>timestamp without time zone</i> y <i>timestamp with time zone</i> .

6.2.12. fake_data_gen_test

Identificador	PU-71
Descripción	Se intenta formatear un dato generado aleatoriamente.
Resultado esperado	El dato es envuelto en comillas simples.

Identificador	PU-72
Descripción	Se intenta formatear un intervalo de 23 días y 45 segundos.
Resultado esperado	23 d 45 s

Identificador	PU-73
Descripción	Se intenta generar una cadena de caracteres. Se intenta generar una cadena de caracteres de longitud máxima 23.
Resultado esperado	La cadena generada por defecto tiene menos de 11 caracteres. La cadena generada con longitud máxima no supera esa longitud. No se generan cadenas vacías.

Identificador	PU-74
Descripción	Se intentan generar un número entero de 2 bytes y otro de 4.
Resultado esperado	Ambos números se encuentran dentro de sus límites.

Identificador	PU-75
Descripción	Se intenta generar un valor de verdad.
Resultado esperado	El valor generado es verdadero o falso. El valor es de tipo <i>bool</i> .

Identificador	PU-76
Descripción	Se intenta generar un número decimal con límites establecidos.
Resultado esperado	El número decimal generado se encuentra dentro de los límites.

Identificador	PU-77
Descripción	Se intenta generar una fecha.
Resultado esperado	La fecha se corresponde con el tipo <i>datetime</i> .

Identificador	PU-78
Descripción	Se intenta generar un intervalo.
Resultado esperado	El intervalo generado se corresponde con el tipo <i>timedelta</i> .

Identificador	PU-79
Descripción	Se intenta generar un dato único de un conjunto de datos que ya se han utilizado.
Resultado esperado	El dato generado es único.

Identificador	PU-80
Descripción	Se intenta generar un dato para una columna que es clave foránea.
Resultado esperado	El dato obtenido forma parte de los datos ya generados en la tabla referenciada.

Identificador	PU-81
Descripción	Se intenta generar un dato para cada tipo soportado.
Resultado esperado	La generación se ejecuta correctamente.

Identificador	PU-82
Descripción	Se intentan generar datos con tipos personalizados.
Resultado esperado	Los datos son generados correctamente.

Identificador	PU-83
Descripción	Se intenta generar una columna con un tipo de dato no soportado.
Resultado esperado	Se produce una excepción de tipo <i>InvalidColumnTypeError</i> .

Identificador	PU-84
Descripción	Se intentan generar datos en grupos más grandes.
Resultado esperado	Los datos generados se producen en las cantidades especificadas.

Identificador	PU-85
Descripción	Se intenta generar un dato en una columna <i>NULLABLE</i> .
Resultado esperado	Aparece <i>NULL</i> entre los datos generados.

Identificador	PU-86
Descripción	Se intenta hacer una generación de datos muy restrictiva.
Resultado esperado	La generación en bloque se renueva varias veces hasta generar todos los datos.

Identificador	PU-87
Descripción	Se intenta obtener un grupo de datos a partir de los presentes en la columna referenciada.
Resultado esperado	El grupo de datos está formado por los datos de la columna referenciada.

Identificador	PU-88
Descripción	Se intentan generar datos para una columna que referencia a otra.
Resultado esperado	Los datos generados se corresponden con datos existentes en la columna referenciada.

Identificador	PU-89
Descripción	Se intenta iniciar el generador de datos para una tabla concreta.
Resultado esperado	Se inicia correctamente un objeto <i>Faker</i> . El objeto generado se corresponde con la configuración de la tabla.

Identificador	PU-90
Descripción	Se intenta iniciar el generador de datos a partir de una tabla que no está presente en el contexto actual.
Resultado esperado	Se produce una excepción de tipo <i>TableNotInContext</i> .

Identificador	PU-91
Descripción	Se intenta generar un dato usando un conjunto de datos generados con un único elemento.
Resultado esperado	El elemento generado se corresponde con el único elemento en el conjunto preparado.

Identificador	PU-92
Descripción	Se intenta detectar una clave primaria válida y una no válida.
Resultado esperado	La tupla no válida es detectada correctamente. La tupla válida es detectada correctamente.

Identificador	PU-93
Descripción	Se comprueba si quedan datos únicos en conjuntos que los han agotado.
Resultado esperado	Se detectan correctamente los conjuntos sin valores únicos.

Identificador	PU-94
Descripción	Se comprueba si un dato duplicado ya ha sido generado.
Resultado esperado	Se detecta correctamente que el dato ya existe en una columna.

Identificador	PU-95
Descripción	Se intentan recuperar los valores generados para una tupla.
Resultado esperado	Los valores se obtienen siempre en el mismo orden. Los valores se recuperan correctamente.

Identificador	PU-96
Descripción	Se modifica una columna a un tipo no soportado y luego se intenta generar datos para esa columna.
Resultado esperado	Se produce una excepción de tipo <i>InvalidColumnTypeError</i> .

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusión y trabajo futuro

7.1.1. Objetivos alcanzados

Se han conseguido lograr los siguientes objetivos:

- Se han conseguido desarrollar los requisitos recogidos.
- Se ha conseguido aprender las nuevas tecnologías durante la realización del proyecto.
- Se ha profundizado y consolidado el conocimiento en varias áreas vistas en la carrera.
- Se ha conseguido terminar el proyecto.

Creo conveniente destacar el último punto. Terminar un proyecto ambicioso me parece una buena forma de terminar la carrera ya que una de las mejores lecciones que se pueden aprender es cómo terminar aquello que se empieza. Muchos de los proyectos que se emprenden en el mundo real no llegan a terminar, por lo que considero que ser capaz de terminar un proyecto emulando un contexto profesional es una de las mejores cartas de presentación que se pueden tener.

7.1.2. Objetivos no alcanzados

No se ha conseguido terminar el proyecto en el plazo estimado. Para una buena gestión de proyecto se necesita bastante experiencia. En la parte de las estimaciones, esta falta de experiencia ha penalizado bastante de cara a la organización de tareas. Esto sumado a las situaciones personales, ha dilatado el proyecto considerablemente.

A pesar de todo esto, el límite temporal era conseguir terminar el proyecto antes de octubre de 2020. Aunque apurado, se ha conseguido terminar dentro de la cantidad de tiempo disponible.

7.1.3. Líneas de trabajo futuro

La aplicación tiene unas funcionalidades básicas sólidas que pueden mejorarse en múltiples aspectos.

Eficiencia

Se observó que, en un intento por mejorar la eficiencia de la aplicación, se veía afectada también la distribución aleatoria de los datos generados. Esto podría trabajarse más en profundidad para permitir una rápida generación de datos y una mejora en la eficiencia de la aplicación. Esta mejora permitiría poder rellenar bases de datos más complejas.

Alcance

La aplicación da soporte a un subconjunto del estándar SQL-92. En concreto, es dependiente del sistema gestor de base de datos Postgresql. Otra línea de trabajo consistiría en aumentar la cantidad de elementos dentro del estándar a la que se da soporte. También se puede trabajar para hacer que el sistema sea independiente del tipo de sistema gestor que se use.

Web

Se pueden hacer aplicaciones web que usen el núcleo de la aplicación. Se podrían implementar sistemas de usuarios, escenarios compartidos o métodos de gamificación para hacer la aplicación más efectiva en su aspecto didáctico. En la parte del servidor se podría adaptar el núcleo de la aplicación para gestionar sesiones concurrentes y poder procesar a varios usuarios asegurando la estabilidad del sistema.

7.1.4. Opinión personal

Este proyecto ha supuesto un gran reto para mí. Resume perfectamente lo que significa cursar una carrera universitaria, ya que supone combinar el conocimiento transmitido por los profesores con la búsqueda de información externa para asimilar un conocimiento propio basado en el contraste de ideas.

He aprendido durante el proyecto. No solo en conocimientos técnicos, sino en otro tipo de habilidades transversales que son en mi opinión bastante más valiosas. He cambiado mi forma de organizarme varias veces durante este recorrido y he probado varias metodologías y diferentes maneras de adaptar mi estilo de trabajo. Este trabajo me ha preparado para el mundo laboral de una forma que no esperaba. No puedo pedir un mejor final para este recorrido académico.

Apéndice A

Manual de instalación

A.1. Requisitos

El ejecutable de la aplicación viene con las librerías de PyQt5 para la interfaz gráfica y un intérprete de Python incluido. Solo será necesario instalar Postgresql en una versión superior a 9.5 en el caso de que se quiera tener la base de datos en la máquina local.

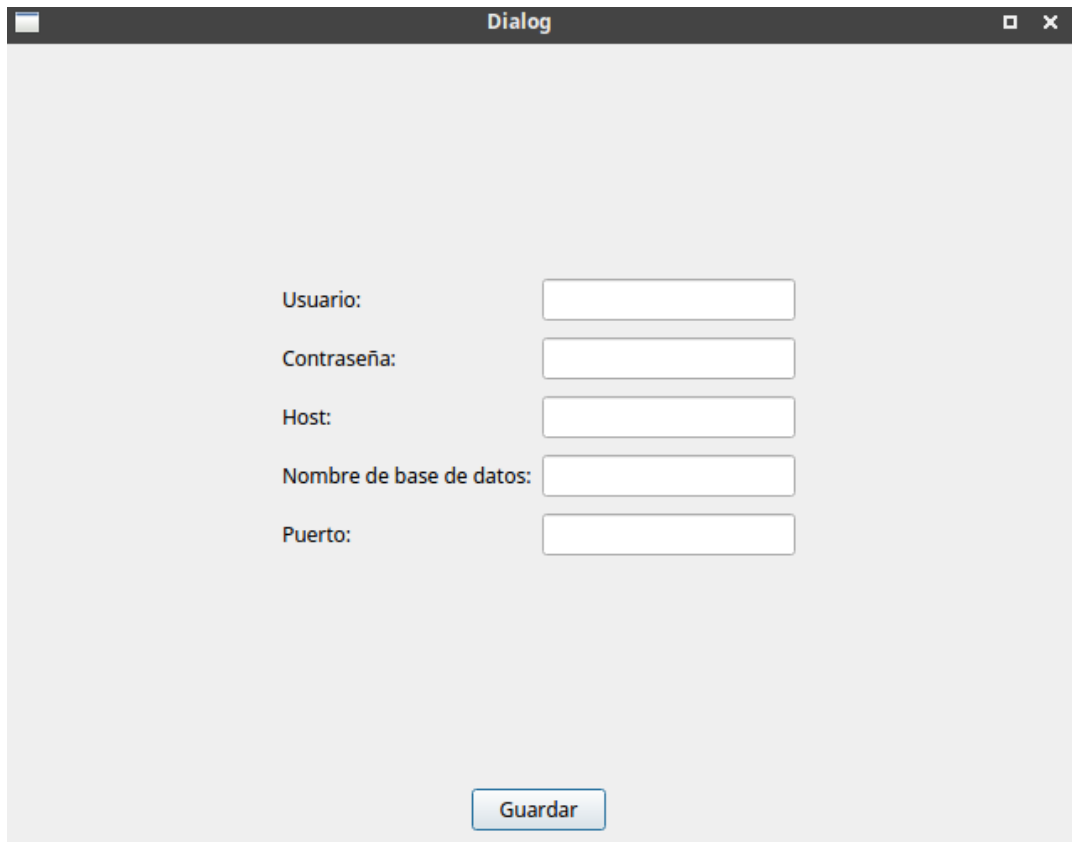
Para poder conectarse con Postgresql es necesario cumplir lo siguiente:

- La versión de Postgresql debe ser superior a 9.5.
- El usuario con el que nos conectemos debe tener permisos elevados.
- Debemos conocer el *host*, o dirección en la que está alojado Postgresql.
- Se debe conocer el puerto que expone Postgresql. Por defecto, 5432.
- Se deben conocer las credenciales del usuario. Nombre y contraseña.
- Una base de datos creada en la que alojar la información de la aplicación.

A.2. Instrucciones de instalación

El ejecutable contiene todos los archivos necesarios para funcionar, solo requiere de una breve configuración durante la primera ejecución.

Al abrir la aplicación se nos mostrará la siguiente pantalla:



The image shows a window titled "Dialog" with a standard operating system title bar (minimize, maximize, close buttons). The window contains a configuration form with five input fields, each preceded by a label:

- Usuario:
- Contraseña:
- Host:
- Nombre de base de datos:
- Puerto:

At the bottom center of the dialog is a button labeled "Guardar".

Figura A.1: Instalación paso 1

Tendremos que introducir la información requerida en cada campo. Una vez tengamos todos los datos, hacemos click en el botón *Guardar*. Si todo ha salido bien, se mostrará la pantalla principal de la aplicación.

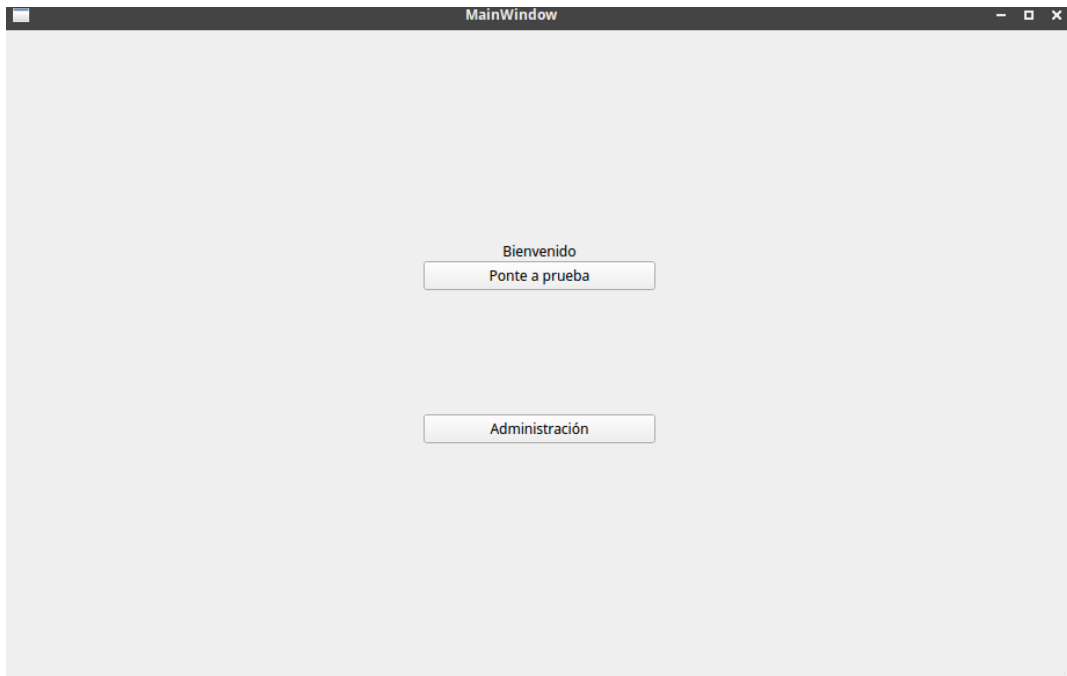


Figura A.2: Instalación paso 2

Apéndice B

Manual de usuario

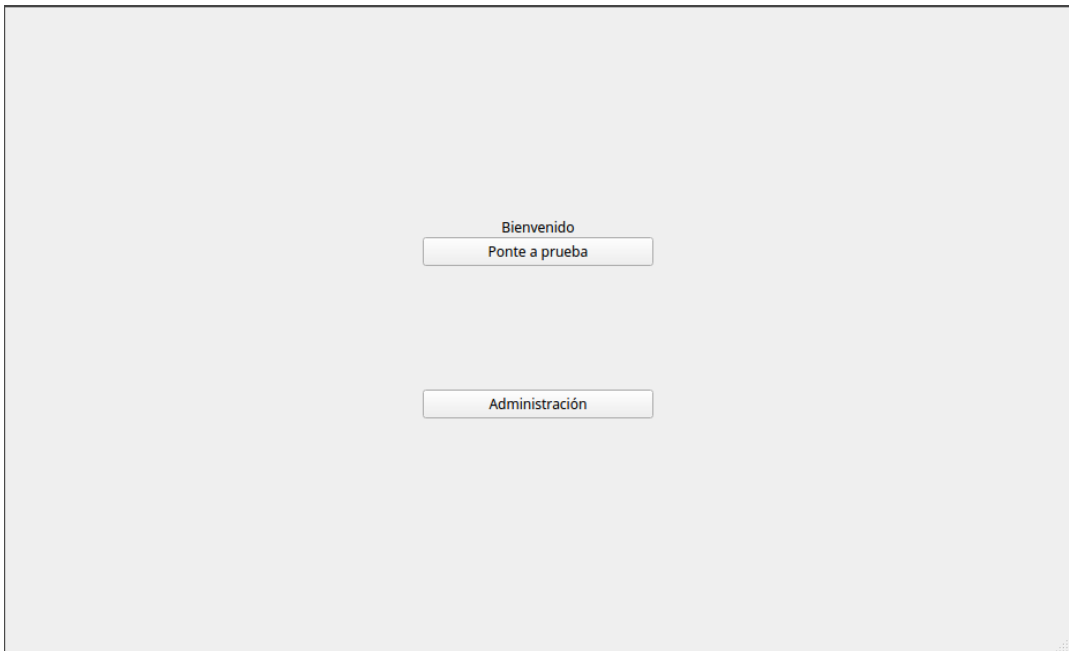
B.1. Menú principal

Al iniciar la aplicación aparecerá el menú principal. Si no es así, la aplicación no está configurada correctamente y se deberá seguir el manual de instalación para hacerlo.

Desde este menú, podemos acceder a las dos partes principales de la aplicación.

Presionando el botón *Ponte a prueba*, iremos al menú de *selección de escenarios*, donde podremos contestar preguntas y obtener informes dependiendo de nuestro desempeño.

Presionando el botón *Administración* iremos al menú de administración, donde podremos configurar escenarios y tipos de datos personalizados.



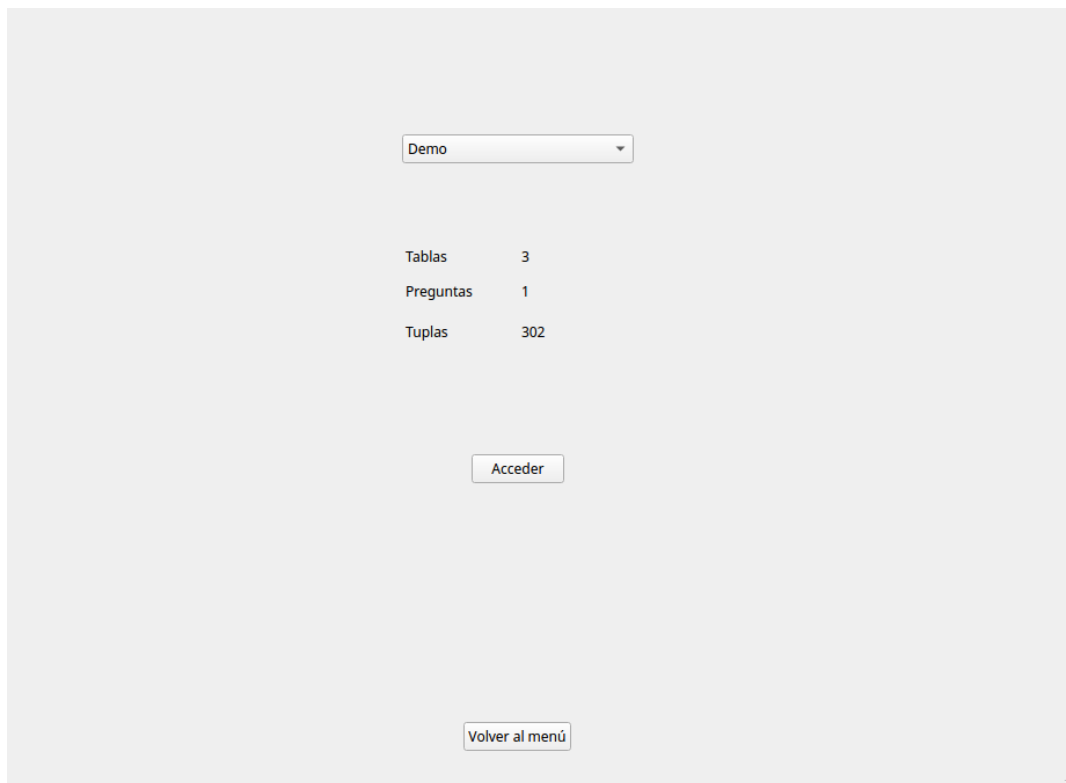
B.2. Menú de escenarios disponibles

En este menú se muestra una previsualización del escenario seleccionado. Podemos seleccionar nuevos escenarios presionando sobre el desplegable y haciendo click en el escenario que deseemos.

La previsualización muestra algunos datos sobre cada escenario. El número de tablas que hay en la base de datos de ese escenario, la cantidad de preguntas que vamos a tener que responder y la cantidad de tuplas presentes en las tablas de esa base de datos.

Para comenzar a responder las preguntas del escenario seleccionado, presionaremos sobre el botón *Acceder*.

Para volver al menú principal presionaremos el botón *Volver al menú*.



B.3. Resolver preguntas

En la parte izquierda de la pantalla veremos un listado de preguntas numeradas. Podemos presionar en cada una para navegar entre ellas. Según vayamos cambiando de pregunta, el enunciado de la pregunta que se muestra en la parte superior se irá actualizando.

The screenshot shows a web application interface for solving questions. On the left side, there is a vertical list of questions, with the first one labeled "Pregunta 0" highlighted in blue. The main area on the right is titled "Pregunta: Nombres de las personas cuyo DNI contiene la letra A". Below the title is a large empty rectangular box, likely for displaying the question's content or results. Underneath this box, the text "Introduce la consulta:" is followed by a "Probar consulta" button. Below the "Introduce la consulta:" text is another large empty rectangular box, intended for the user to enter a query. At the bottom of the interface, there are two buttons: "Salir" on the left and "Terminar y corregir preguntas" on the right.

El campo de texto en la parte inferior de la pantalla será donde introduzcamos la respuesta a nuestra consulta. Podemos presionar sobre el botón *Probar consulta* para comprobar que nos devuelve de resultado la base de datos. El resultado de la consulta aparecerá en el cuadro de la parte superior.

Pregunta: Nombres de las personas cuyo DNI contiene la letra A

```
-313088137 HUGO MARTIN B33634555 1151-03-15 True m
-1917214306 FEDERICO RAMON O34397852 0911-06-23 True f
1453119062 FRANCISCO ROQUE D89698325 2000-11-26 True f
-620653136 ANGEL ABEL J79381289 1331-09-17 False m
-2005985934 EUGENIO JOAQUIN F10545258 1834-01-01 True f
-1921214620 CARLES MARIA T64841091 0620-11-25 True f
-1254920655 CHARKI M35365178 1788-10-13 True f
-1737581857 FRANCESC ANTONI G72234664 0153-04-21 True f
1874032047 AXULAR J13223942 1331-09-17 False f
1018631174 EUGENIO JOAQUIN U18495600 0153-04-21 True m
738603651 SERGIO ARTURO A36622615 1371-01-18 True f
```

Introduce la consulta:

select * from persona;

Si nuestra consulta contiene algún error grave, se nos informará en el cuadro de la parte superior. Para salir sin recibir un informe de corrección, presionaremos el botón *Salir*. Esto nos llevará de vuelta al menú anterior.

Para corregir las preguntas que hayamos contestado, presionaremos el botón *Terminar y corregir preguntas*. Esto nos llevará a la pantalla con el informe de resultados. Una vez allí, no podremos volver a responder las preguntas desde el punto en el que se dejó al corregir. Si se quiere volver a contestar las preguntas se tendría que volver a empezar desde cero.

Podemos volver a preguntas ya contestadas para modificar la respuesta y asegurarnos de que nos devuelve los resultados que queramos. Para dejar una pregunta sin contestar deberemos asegurarnos de que se haya eliminado todo el texto en el cuadro inferior.

Pregunta 0

Pregunta: Nombres de las personas cuyo DNI contiene la letra A

-620653136 ANGEL ABEL J79381289 1331-09-17 False m
-2005985934 EUGENIO JOAQUIN F10545258 1834-01-01 True f
-1921214620 CARLES MARIA T64841091 0620-11-25 True f
-1254920655 CHARKI M35365178 1788-10-13 True f
-1737581857 FRANCESC ANTONI G72234664 0153-04-21 True f
1874032047 AXULAR J13223942 1331-09-17 False f
1018631174 EUGENIO JOAQUIN U18495600 0153-04-21 True m
738603651 SERGIO ARTURO A36622615 1371-01-18 True f
Error sintáctico detectado, revisa tu consulta
Error sintáctico detectado, revisa tu consulta
Error sintáctico detectado, revisa tu consulta

Introduce la consulta: Probar consulta

consulta incorrecta

Salir
Terminar y corregir preguntas

B.4. Informe de corrección

En esta pantalla se muestra un informe detallado de los resultados en cada pregunta. Se muestra un conteo de preguntas acertadas respecto al total en la parte superior izquierda.

Podremos navegar por las preguntas de forma análoga a cuando se contestan las preguntas, presionando sobre la pregunta que queramos ver el informe detallado.

De cada pregunta sabremos si está resuelta correctamente; es decir, si cumple todos los criterios para darse por buena respuesta. Sabremos si el resultado de la consulta es el correcto, o lo que es lo mismo, si devuelve el mismo resultado que una respuesta correcta. Se nos mostrará un listado de las tablas que hayamos usado y no se consideraran necesarias para responder a la pregunta. También se nos mostrará un listado de las palabras clave necesarias o prohibidas y si hemos cumplido esas restricciones.

Si presionamos el botón *Volver al menú*, volveremos al menú de selección de escenario. Una vez allí, podremos realizar más pruebas si así lo deseamos.

0 de 1 preguntas contestadas correctamente

Pregunta 0

Pregunta: Nombres de las personas cuyo DNI contiene la letra A

Pregunta resuelta correctamente: **No**

Resultado de la consulta: **Incorrecto**

Tablas innecesarias usadas

Requisitos de palabras clave

- Prohibido: BY ✓
- Prohibido: GROUP ✓

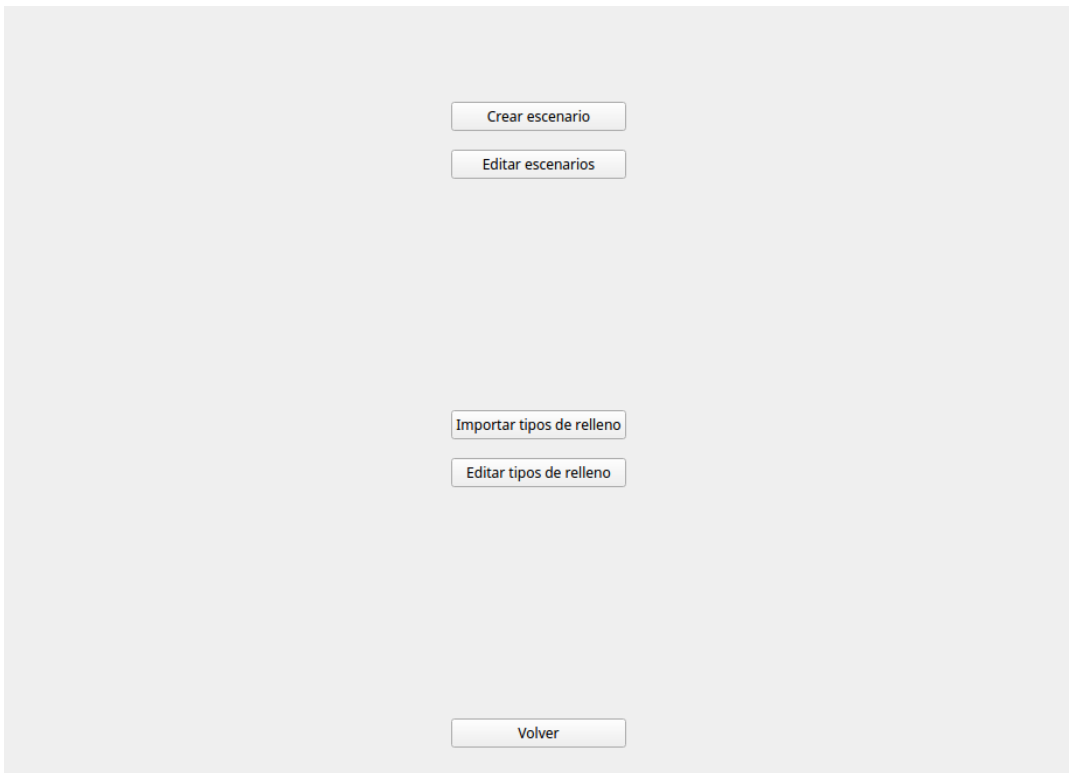
Volver al menú

B.5. Menú de administración

En este menú se recogen las principales funciones de la administración del sistema:

- Para crear escenarios pulsaremos sobre el botón *Crear escenario*.
- Para editar los escenarios existentes pulsaremos el botón *Editar escenarios*.
- Para cargar en la aplicación nuevos tipos de datos de relleno, pulsaremos el botón *Importar tipos de relleno*.
- Para editar los tipos de datos de relleno existentes pulsaremos el botón *Editar tipos de relleno*.

Para volver de vuelta al menú principal, pulsaremos sobre el botón *Volver*.



B.6. Creación de escenarios

En esta pantalla seguiremos un proceso de 3 pasos para crear un escenario. Primero crearemos un escenario a partir del esquema de su base de datos. Luego configuraremos los

datos que queremos generar. Finalmente, añadiremos las preguntas que tengamos preparadas y terminaremos el proceso.

Una vez terminado el proceso se generarán los datos necesarios del escenario.

Si el proceso se interrumpe después de haber avanzado al paso de generar datos, se podrá continuar preparando el escenario desde el menú de edición.

B.6.1. Esquema

El primer paso de creación de un escenario consiste en ponerle un nombre y cargar la definición de la base de datos.

Esquema Generar datos Configurar preguntas

Paso 1 elige el nombre del escenario e introduce el DDL:

Nombre del escenario:

Carga la definición de las tablas desde un archivo:

Cargar archivo Ningún archivo seleccionado

Puedes introducir o modificar la definición manualmente en este área de texto:

Confirmar esquema Volver al menú

Una vez introducido un nombre de escenario y haber definido la base de datos, pulsaremos sobre el botón *Confirmar esquema*. Después de confirmar el esquema, el sistema creará la configuración inicial del escenario y si nada sale mal, se habilitará el botón *Generar datos* en la barra de navegación superior.

Podremos volver al menú de administración en cualquier momento pulsando sobre el botón *Volver al menú*.

Esquema ▶ Generar datos ▶ Configurar preguntas

Paso 1 elige el nombre del escenario e introduce el DDL:

Nombre del escenario:

Carga la definición de las tablas desde un archivo:

Cargar archivo

Puedes introducir o modificar la definición manualmente en este área de texto:

```
CREATE TABLE Prestamo (
  fecha TIMESTAMPTZ,
  fecha_local TIME,
  duracion_maxima INTERVAL,
  titulo VARCHAR,
  autor VARCHAR,
  persona INT,
  PRIMARY KEY (titulo, autor, persona),
  FOREIGN KEY (persona) REFERENCES persona (id)
```

B.6.2. Datos

En esta pantalla configuraremos la generación de datos de la base de datos que hemos creado.

Podemos seleccionar la tabla que queramos configurar presionando sobre las pestañas de la parte superior. Debajo de la barra de pestañas podemos configurar la cantidad de datos que se generarán en cada tabla.

En el centro de la pantalla se nos mostrará un listado de las columnas de la tabla seleccionada. Podemos configurar el tipo de dato que se generará para cada columna siempre y cuando sea compatible con el tipo definido en la definición del paso anterior. Para cada columna tenemos 3 modos diferentes de generación:

- Expresiones regulares seleccionando *Regex*.
- Aleatorio, el modo por defecto.
- Datos personalizados seleccionando *Tipos personalizados*.

En el modo de expresiones regulares es obligatorio rellenar tanto el campo principal, con nuestra expresión regular, como el campo de longitud máxima.

En el modo aleatorio todos los campos son opcionales. Hay que tener en cuenta que, dependiendo de la columna, los valores máximo y mínimo cambiarán de formato. Si la columna representa fechas, tendremos que introducir fechas; si es un número, en cambio, tendremos que utilizar números. La longitud máxima no tendrá ningún efecto en la generación de datos con un tipo que no soporte longitudes máximas.

En el modo de datos personalizados podremos elegir el tipo de datos entre una lista que aparecerá al pulsar sobre el desplegable. Si el desplegable está vacío no podremos usar este modo.

Esquema ▶ Generar datos ▶ Configurar preguntas

Paso 2 - Configura los datos para rellenar cada columna

libro persona prestamo

Cantidad de datos a generar:

Columna	Tipo	Configuración	Longitud máxima
ejemplares	<input type="radio"/> Regex	Expresión regular	<input type="text"/>
	<input checked="" type="radio"/> Aleatorio	Valor máximo <input type="text"/> Valor mínimo <input type="text"/>	<input type="text"/>
	<input type="radio"/> Tipo personalizado	Titles	
titulo	<input checked="" type="radio"/> Regex	[A-Z]+	<input type="text"/>
	<input type="radio"/> Aleatorio	Valor máximo <input type="text"/> Valor mínimo <input type="text"/>	<input type="text"/>
	<input type="radio"/> Tipo personalizado	Titles	
autor	<input type="radio"/> Regex	Expresión regular	<input type="text"/>
	<input type="radio"/> Aleatorio	Valor máximo <input type="text"/> Valor mínimo <input type="text"/>	<input type="text"/>
	<input checked="" type="radio"/> Tipo personalizado	nombres_masculinos	

[Volver al menú](#)

B.6.3. Preguntas

En esta pantalla podremos configurar las preguntas del escenario. En el cuadro de la izquierda se escribe el enunciado del problema o la pregunta a resolver. En el cuadro de la derecha se escribe la consulta considerada como correcta para esa pregunta. Una vez descrita tanto pregunta como respuesta pulsaremos el botón *Añadir pregunta*.

Las preguntas que hayan sido añadidas se mostrarán en el cuadro central, pero no se guardarán hasta terminar el escenario. Podemos borrar las preguntas añadidas que queramos pulsando sobre *Borrar pregunta*.

Podemos configurar las palabras clave de una pregunta pulsando sobre el botón *Palabras clave*. Esto nos abrirá una pantalla emergente en la que podremos configurar las palabras clave de la pregunta seleccionada.

Esquema ► Generar datos ► Configurar preguntas Terminar

Paso 3 - Añade las preguntas

Todos la información disponible sobre libros select * from libro; Palabras clave Borrar pregunta

Pregunta	Consulta correcta
Todos la información disponible sobre libros	select * from libro;

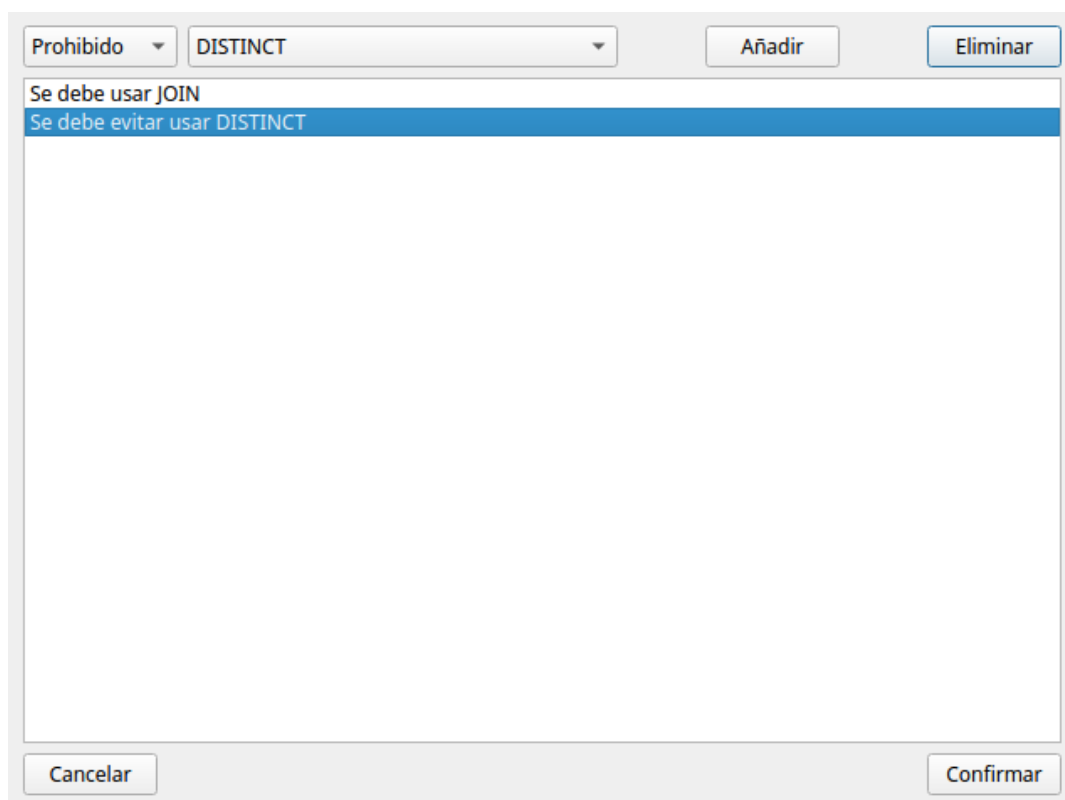
Añadir pregunta Volver al menú

Para añadir palabras clave seleccionaremos el estado que queremos para la palabra clave:

- *Obligatorio*, si queremos que se use esa palabra clave.
- *Prohibido*, si queremos penalizar el uso de esa palabra clave.

Seleccionaremos la palabra clave de entre las que se dispongan en el desplegable.

Una vez ambos campos estén configurados como queremos, haremos click en el botón *Añadir*. Podremos ver las palabras clave añadidas en el cuadro central de la ventana. Si queremos borrar alguna palabra clave, seleccionaremos la deseada y pulsaremos sobre el botón *Eliminar*.



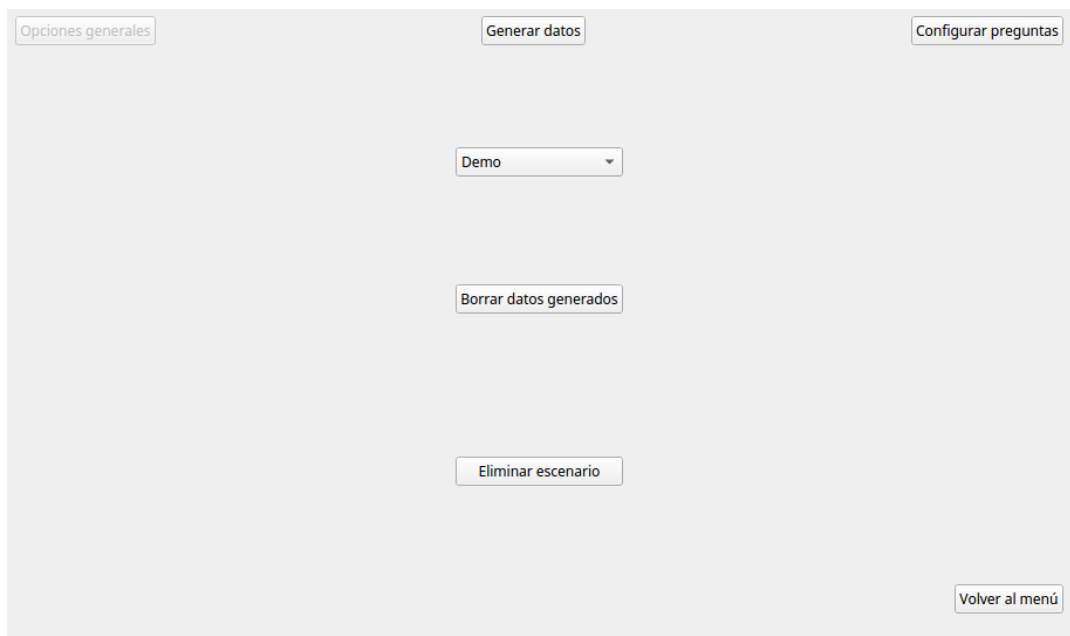
B.7. Edición de escenarios

Esta pantalla permite realizar ligeras modificaciones a un escenario. Podremos navegar entre las posibles categorías de edición con los botones de la parte superior de la pantalla. Cada botón nos llevará a su respectiva pantalla de edición. Podemos saber en qué pantalla nos encontramos fijándonos en el botón que se encuentra deshabilitado.

Podremos volver al menú de administración en cualquier momento pulsando sobre el botón *Volver al menú*.

B.7.1. Configuración general

Aquí podremos vaciar el contenido de la base de datos de un escenario o borrar el escenario completamente. Esto lo podremos hacer pulsando sobre los botones *Borrar datos generados* y *Eliminar escenario* respectivamente.



The screenshot shows a web interface for scenario configuration. At the top, there are three buttons: 'Opciones generales' (disabled), 'Generar datos', and 'Configurar preguntas'. Below these is a dropdown menu currently set to 'Demo'. Further down are two buttons: 'Borrar datos generados' and 'Eliminar escenario'. At the bottom right corner, there is a button labeled 'Volver al menú'.

B.7.2. Generar datos

Aquí podremos generar datos adicionales de forma similar a la que seguíamos en el menú de creación de escenario. La única salvedad es que tendremos que pulsar el botón *Generar datos* que se encuentra habilitado. Cada vez que pulsemos se generarán datos adicionales según la configuración que hayamos elegido.

The screenshot shows a web interface for generating data. At the top, there are three buttons: 'Configuración general', 'Generar datos', and 'Configurar preguntas'. Below this, a dropdown menu shows 'Selección escenario: Demo'. There are two 'Generar datos' buttons. Below that, there are three tabs: 'libro', 'persona', and 'prestamo'. A text input field shows 'Cantidad de datos a generar: 100'. The main area is a scrollable list of field configurations. Two fields are visible: 'titulo' and 'precio'. Each field has three radio button options: 'Regex', 'Aleatorio', and 'Tipo personalizado'. The 'Aleatorio' option is selected for both. For 'titulo', the 'Aleatorio' option has sub-inputs for 'Valor máximo', 'Valor mínimo', and 'Longitud máxima'. The 'Tipo personalizado' option has a dropdown menu showing 'Titles'. The 'precio' field also has sub-inputs for 'Valor máximo', 'Valor mínimo', and 'Longitud máxima'. At the bottom right, there is a 'Volver al menú' button.

B.7.3. Configurar preguntas

Aquí podremos añadir y eliminar preguntas, así como editar sus respectivas palabras clave. La forma de hacerlo es igual que en el proceso de creación de escenario. Como única salvedad, para guardar las preguntas y que se generen los datos adicionales de las preguntas, deberemos pulsar sobre el botón *Guardar preguntas*.

Configuración general Generar datos Configurar preguntas

Seleccione escenario: Demo Guardar preguntas

Nombres de las personas cuyo DNI contiene la letra select nombre from persona where dni like 'A%'; Palabras clave Borrar pregunta

Pregunta

Consulta correcta

Añadir pregunta Volver al menú

B.8. Importación de tipos personalizados

En esta pantalla podremos cargar tipos personalizados para usar cuando generemos datos para los escenarios. Podemos cargar tipos personalizados desde un archivo CSV y se cargarán como cualquier editor de hojas de cálculo. La primera fila se usará como nombre de los tipos y se añadirá el texto en la cabecera de las columnas.

También podemos añadir tipos a mano, haciendo doble click sobre las celdas para escribir en ellas. Podemos añadir filas y columnas usando los respectivos botones *Añadir fila* y *Añadir columna*.

Cargar un archivo CSV sobrescribirá todas las celdas. Es mejor añadir datos adicionales después de haber cargado el archivo.

Podremos eliminar las celdas que queramos. Para ello, seleccionaremos las celdas deseadas y pulsaremos sobre el botón *Eliminar selección*.

Para volver al menú de administración, pulsaremos el botón *Volver al menú*.

The screenshot shows a web interface for managing custom types. At the top left, there is a button labeled "Cargar desde archivo" and a status indicator "Ningún archivo seleccionado". At the top right, there is a button labeled "Guardar tipos". Below these, there are two buttons: "Añadir fila" and "Eliminar selección". On the left side, there is a button labeled "Añadir columna". The main area is a large empty table. At the bottom right, there is a button labeled "Volver al menú".

Cargar desde archivo short_type.csv Guardar tipos

Añadir fila Eliminar selección

Añadir columna	Short
1	1
2	2
3	3
4	4
5	5

Volver al menú

B.9. Edición de tipos personalizados

Aquí podremos ver todos los tipos disponibles en la aplicación y editarlos de forma similar a la pantalla de importación. Para guardar las modificaciones que hayamos hecho, pulsaremos el botón *Guardar tipos*.

The screenshot shows a web application interface for editing custom types. At the top right, there is a button labeled "Guardar tipos". Below it, on the right side, is a button labeled "Eliminar selección". The main area contains a table with the following data:

nombres_masculinos	
1	JOSE
2	MANUEL
3	FRANCISCO
4	JUAN
5	DAVID
6	JOSE ANTONIO
7	JOSE LUIS
8	JAVIER
9	JESUS
10	FRANCISCO JAVIER
11	CARLOS
12	DANIEL
13	MIGUEL
14	RAFAEL
15	JOSE MANUEL
16	PEDRO

At the bottom right of the interface, there is a button labeled "Volver al menú".

Apéndice C

Manual de mantenimiento

C.1. General

La aplicación se divide en dos grandes partes. Una controla la interfaz y la otra, gran parte de la lógica de negocio de la aplicación. Las modificaciones que requieran modificar la lógica de negocio pueden realizarse modificando los controladores de la interfaz, siempre y cuando la modificación pertenezca de funcionalidad presente en la aplicación. En el caso de querer ampliar las capacidades de la aplicación o añadir nuevas funcionalidades, se deberá trabajar en el núcleo de la aplicación.

C.2. Núcleo de la aplicación

Dependiendo de las modificaciones que se quieran realizar, se modificarán unas clases u otras. Como regla general, se deberán realizar los cambios en las clases que sean responsables del comportamiento que se quiera incluir o modificar. Esta lógica también se puede aplicar para encontrar funcionalidad que se quiera eliminar.

Para cualquier modificación que se realice, se recomienda la ejecución de los tests unitarios con la herramienta Pytest. Cualquier modificación que suponga cambiar el comportamiento esperado requerirá modificar aquellos tests que prueben esa parte de la funcionalidad.

La comunicación y gestión de la base de datos es responsabilidad del módulo *connection_manager*.

El rellenado de la base de datos es responsabilidad de la clase *filler*.

La generación de datos es responsabilidad del módulo *fake_data_gen*.

La administración de tipos personalizados es responsabilidad del módulo *custom_fakes*.

El control de tipos soportados es responsabilidad de *type_compatible*. Si se quiere dar soporte a nuevos tipos de datos o modificar el nombre de los existentes, se debe cambiar aquí. Esto puede ser necesario para dar soporte a otros gestores de bases de datos, o incluir algún tipo de PostgreSQL que no se encuentre ya soportado.

La configuración de la generación de datos es responsabilidad del módulo *structures*. Este módulo agrupa gran parte del modelo de dominio. Las modificaciones que impliquen mayor personalización durante la generación de datos, pasarán por aquí. También será necesario modificar este módulo si se quieren obtener datos adicionales de la base de datos junto con *schema_recollector*.

La conversión del DDL al dominio de la aplicación es responsabilidad de la clase *schema_recollector*.

La definición de consultas predeterminadas que se vayan a utilizar se guardan en el archivo *queries.py* dentro del módulo *utils*. El módulo *utils* almacena definiciones de clases e interfaces no nativas en Python en el archivo *metaclasses.py*.

La lógica de corrección es responsabilidad del módulo *judge*. Para modificar la gestión de una prueba o los datos que incluye un informe de resultados, se deberá modificar este módulo.

El análisis previo de las preguntas, al momento de ser creadas, es responsabilidad del módulo *questions*. También es responsabilidad de este módulo la generación adicional de datos para aumentar la precisión del juez.

C.3. Interfaz

La aplicación sigue el patrón MVC (Modelo Vista Controlador). Los cambios que se quieran realizar deberán seguir este patrón.

Los cambios en la lógica deberán estar definidos en un controlador.

Los cambios en la forma de visualización deberán estar definidos en su correspondiente vista.

Los cambios en la forma de almacenar los datos o los datos que se quieren guardar, deberán estar definidos en su correspondiente modelo.

Para modificaciones ligeras de las vistas se pueden usar los archivos `.ui` almacenados en `view/qt_view`. Estos archivos se pueden editar con la aplicación QtDesigner y luego ser transformados en archivos `.py` mediante el programa `pyuic`. La aplicación usa PyQt5 y se necesitará la versión equivalente de QtDesigner y `pyuic`.

Es importante no hacer modificaciones dentro de los archivos `.py` dentro de `view/qt_view`. Estos archivos son autogenerados y la aplicación los utiliza mediante herencia. Si se quieren añadir vistas adicionales, se deberá seguir un proceso similar: Primero se genera un archivo `.ui` y se compila en su versión `.py` equivalente. Después se creará una vista fuera del módulo `qt_view` y se heredará el archivo autogenerado para poder extender las funciones que deseemos.

Bibliografía

- [1] *Sql-ex*,
SQL-EX.RU.
Recuperado el 20/09/2020,
de <https://sql-ex.ru/>
- [2] *Sqlfiddle*,
SQLFIDDLE.COM.
Recuperado el 20/09/2020,
de <http://sqlfiddle.com/>
- [3] *Livesql*,
LIVESQL.ORACLE.COM.
Recuperado el 20/09/2020,
de <https://livesql.oracle.com/>
- [4] *Sqlzoo*,
SQLZOO.NET.
Recuperado el 20/09/2020,
de <https://sqlzoo.net/>
- [5] *Schemaverse*,
SCHEMAVERSE.COM.
Recuperado el 20/09/2020,
de <https://schemaverse.com/>
- [6] *Sql-tutorial*,
SQL-TUTORIAL.RU.
Recuperado el 20/09/2020,
de <http://www.sql-tutorial.ru/>
- [7] *W3Schools*,
W3SCHOOLS.COM.
Recuperado el 20/09/2020,
de <https://www.w3schools.com/>
- [8] *Descriptores de Python*,
DOCS.PYTHON.ORG.

Recuperado el 20/09/2020,
de <https://docs.Python.org/3/howto/descriptor.html>

- [9] *Balsamiq*,
BALSAMIQ.COM.
Recuperado el 20/09/2020,
de <https://balsamiq.com/>