

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

GameSQL

Un tutor de SQL basado en las técnicas de Gamification.

GameSQL

A SQL tutor based on gamification techniques.

Realizado por

Alejandro Fernández Miñambres

Tutorizado por

Manuel Enciso García-Oliveros

Cotutorizado por

Enrique Soler Castillo

Departamento

Lenguajes y Ciencias de la Computación

Universidad de Málaga

Málaga, Junio 2017

Fecha defensa:

El Secretario del Tribunal

Resumen

SQL (*Structured Query Language*) es un lenguaje que fue principalmente diseñado para la manipulación y control de datos en bases de datos. Su funcionamiento principal se basa en el manejo del álgebra y el cálculo relacional, haciendo posible que mediante una serie de consultas podamos recuperar la información.

Por lo tanto, consideramos que su aprendizaje es fundamental para la formación de un alumno de Ingeniería en Informática, Software y Computadores, ya que la mayoría de las aplicaciones hacen uso de algún tipo de base de datos.

Por otro lado, si hablamos de juegos y educación parece que estamos hablando de dos cosas incompatibles. Pero últimamente está en auge el término gamificación (del inglés *gamification*), que consiste en aplicar los conocimientos y mecánicas de los juegos para complementar el aprendizaje.

El funcionamiento principal del cerebro, consiste en probar, experimentar, observar lo que sucede y aprender, funcionamiento que podemos obtener mediante un juego sin el miedo de cometer errores, favoreciendo el grado de satisfacción. Un juego nos permite obtener una relación causa-efecto de manera inmediata, haciendo que aumente el grado de aprendizaje, memorización, etc., además de permitir el poner en práctica lo aprendido.

Con el presente proyecto, **GameSQL: Un tutor de SQL basado en las técnicas de Gamification**, se pretende dar a conocer el lenguaje SQL mediante el uso de los principios que podemos encontrar en los juegos. El alumno podrá probar, experimentar y aprender a base de ensayos y errores, de manera que el proceso de aprendizaje sea más motivador y productivo.

Para ello se desarrollará una aplicación web, donde los alumnos podrán poner en práctica todos los conocimientos aprendidos de manera teórica.

Palabras claves

Juego, SQL, Base de datos, Tutor, Ludificación

Abstract

SQL (Structured Query Language) is a language that was designed for manipulation and control of relational databases. Its operation is based on the handling of algebra and relational calculus, making possible that through a series of queries we can retrieve the information.

Therefore, we consider that their learning is fundamental for the formation of the students of computer engineering, software and computer science, since most of the applications make use of some type of database.

On the other hand, if we talk about games and education it seems that we are talking about two incompatible things, but lately the term gamification is booming, which is to apply the knowledge and mechanics of the game to complement the learning.

The main functioning of the brain, is to try, experiment, observe what happens and learn, functions that we can get in a game without the fear of making mistakes, favoring the degree of satisfaction and obtaining a cause-effect relationship immediately, which increases the level of learning, memorization, etc.

The goal of this project, **GameSQL: A SQL tutor based on the techniques of Gamification**, is to share the SQL language applying the same principles that we can be found in games to achieve a higher level of learning. The student can experiment without risk.

Keywords

Game, SQL, Data base, Tutor, Gamification

Índice

Resumen

Abstract

1	Introducción	3
2	Estado del arte	4
3	Objetivos	6
4	Metodología	7
5	Diseño e implementación	9
5.1	Lenguajes usados	9
5.1.1	Java	9
5.1.2	SQL	10
5.1.3	HTML	10
5.1.4	CSS	10
5.1.5	JavaScript	11
5.1.6	JQuery	11
5.2	Tecnologías utilizadas	11
5.2.1	JSP	11
5.2.2	Apache Tomcat	12
5.2.3	Oracle Database	12
5.3	Entornos de desarrollo utilizados	12
5.3.1	NetBeans IDE	12
5.3.2	DataModeler	12
5.3.3	SQLDeveloper	13
5.4	Framework utilizados	13
5.4.1	Primefaces	13
5.4.2	FontAwesome y Google Fonts	13
6	Funcionalidad requerida	14
6.1	Actores	14
6.1.1	Alumno	14
6.1.2	Profesor	14
6.1.3	Administrador	14
6.2	Requisitos funcionales	14
6.2.1	Requisitos funcionales comunes	14
6.2.2	Requisitos funcionales para el actor “profesor”	14
6.2.3	Requisitos funcionales para el actor “alumno”	16
6.2.4	Requisitos funcionales para el actor “administrador”	17
6.3	Requisitos no funcionales	18
7	Diseño de la aplicación	20

7.1	Casos de uso	20
7.2	Arquitectura	22
7.3	Estructura de la base de datos	23
7.3.1	Modelo de datos conceptual	23
7.3.2	Modelo relacional	24
7.3.3	Modelo o modelos externos	24
7.4	Estructura del proyecto	25
7.5	Sistema de corrección	25
7.5.1	Misma sentencia	26
7.5.2	Comparación de tablas	26
7.6	Seguridad	27
7.6.1	Seguridad mediante permisos	27
7.6.2	Seguridad mediante el control de las entradas	28
7.6.3	Seguridad mediante el control de ejecución	28
7.7	Interfaz	28
7.7.1	Inicio de sesión	30
7.7.2	Interfaz del rol “alumno”	31
7.7.3	Interfaz del rol “profesor”	38
7.7.4	Interfaz del rol “administrador”	45
8	Conclusión	48
	Referencias	49
	Índice de figuras	51
	Anexos	54
	Anexo I: Pruebas unitarias	55
	Anexo II: Manual del alumno	78
	Anexo III: Manual del profesor	86
	Anexo IV: Manual del administrador	95
	Anexo V: Manual de instalación	101

1. Introducción

El aprendizaje de SQL es una tarea fundamental para la formación de un alumno de Ingeniería en Informática, Software y Computadores. Este trabajo, pretende ofrecer una herramienta para el profesorado y alumnado que ayude a adquirir esos conocimientos mediante técnicas de gamificación.

La gamificación (*gamification*) consiste en envolver el aprendizaje en un juego, logrando que sea motivador y ameno, dando la posibilidad de poner en práctica lo aprendido de manera controlada y sin el riesgo de producir errores.

Para ello el alumno contará con un perfil de jugador con diferentes niveles y un sistema de experiencia que irá aumentando según su aprendizaje, además de ponerse a prueba mediante sistemas de evaluación.

Por otro lado, el profesor podrá poner a disposición de los alumnos, un sistema que podrá complementar la parte teoría impartida, además de conocer el estado de aprendizaje de sus alumnos.

2. Estado del arte

El término *gamification* [1] apareció en el mundo anglófono sobre el 2008, pero es en 2010 donde empieza a popularizarse. A día de hoy no hay muchas herramientas web que permitan el aprendizaje del lenguaje SQL en forma de juego.

Actualmente la mayoría de la población tiene acceso a Internet y su expansión es cada día mayor, así como las formas de acceder. Incluso recientemente la ONU declaró que el acceso a Internet es un derecho humano [2]. Por lo tanto, considerando estas características, optamos que la creación de una aplicación web es lo ideal para la creación de este trabajo, ya que nos permite tener un acceso constante, un mantenimiento fácil y un despliegue inmediato.

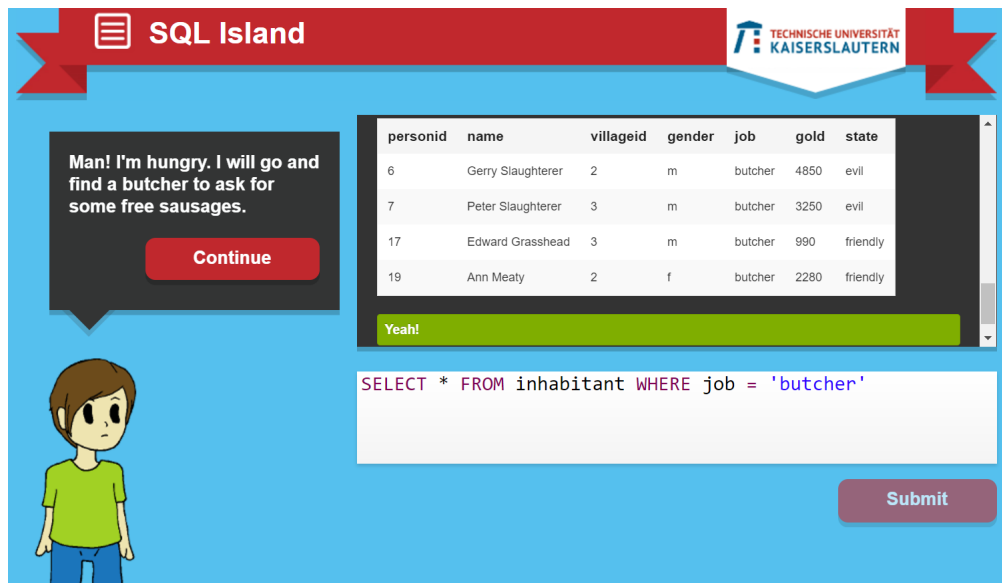


Figura 1: SQL Island.

Tras un análisis exhaustivo en la búsqueda de aplicaciones similares a nuestra propuesta, solo hemos encontrado la aplicación SQL Island [3] la cual, al igual que la nuestra, está enfocada al aprendizaje del lenguaje SQL.

La aplicación está diseñada por la universidad alemana de Kaiserslautern y se trata de una aplicación web que intenta simular una población que habita en una isla, donde mediante la entrada de sentencias en una consola que se proporciona, véase la figura 1, podemos ejecutar diferentes acciones como por ejemplo: consultar el número de habitantes, consultar el salario de estos, simular batallas, etc.

A parte de esta aplicación, también podemos encontrar una gran cantidad de aplicaciones web que funcionan como tutoriales, normalmente guiados, con los cuales vas obteniendo puntos mientras vas avanzando.

Estos tutoriales suelen pertenecer a aplicaciones web que incluyen otros tipos de lenguajes, como Udemy [4], y normalmente son de pago. También los podemos encontrar gratuitos, como SoloLearn [5], Codecademy [6], Codewars [7], etc., pero su funcionamiento no es el mismo.

3. Objetivos

El principal objetivo del trabajo es la creación de una aplicación web fácil e intuitiva con técnicas de gamificación, la cual pueda ayudar y amenizar el aprendizaje de todo lo relacionado con el lenguaje SQL, así como aportar ayuda y simplicidad a la hora de crear pruebas y poder ver el nivel del alumnado. Es decir, crear una herramienta tanto para el alumnado como para el profesorado para simplificar los conocimientos de este lenguaje.

Para garantizar la gamificación, el alumnado contará con un perfil de jugador (avatar, alias, perfil, etc.), así como un sistema de puntos y niveles. La acumulación de puntos hará que el jugador vaya aumentando su nivel, y esta puntuación es obtenida mediante las dos modalidades de juego que son los retos y el entrenamiento.

También dispondrá la posibilidad de abrir canales con el profesor para informar sobre algún error que se produzca, para que el profesor pueda reconsiderar la situación y actuar.

Finalmente el alumno tendrá un apartado donde podrá realizar las diferentes pruebas que el profesorado haya creado y obtener una puntuación, que podrá ser usada para la calificación de la asignatura.

Por parte del profesorado esta herramienta hará que lo enseñado en clase se ponga a prueba con ejemplos reales, garantizando un mejor aprendizaje de sus alumnos. Además, podrá conocer el estado de ellos, crear retos con los alumnos y pruebas que el sistema se encargará automáticamente de corregir, haciendo que se ahorre tiempo.

4. Metodología

Seleccionar una metodología de desarrollo software es uno de los puntos más importantes a la hora de desarrollar un software, porque nos permite estructurar, planear y controlar nuestro desarrollo.

En la actualidad existen varias metodologías de desarrollo software, pero tras analizar los pros y contras de cada una de ellas, nos decantamos por una metodología de desarrollo ágil, en concreto usaremos SCRUM.

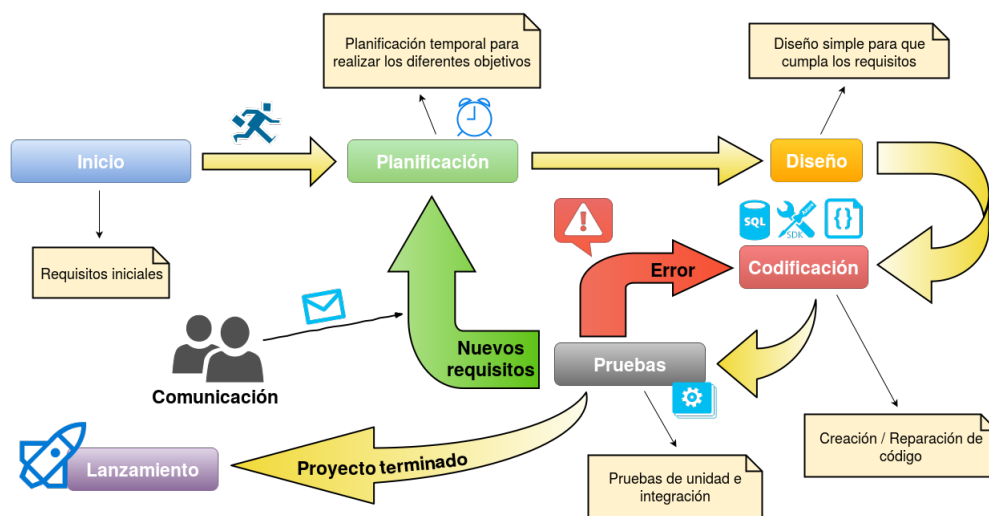


Figura 2: Metodología de desarrollo ágil.

La metodología ágil [8] se basa en la creación y desarrollo incremental del proyecto, donde los requisitos y soluciones evolucionan según las necesidades del proyecto. Por lo tanto, partiendo de la nada, el proyecto ira incrementando poco a poco hasta que se cumplan los requisitos y estos funcionen adecuadamente (figura 2).

Estas metodologías están pensadas para el desarrollo de software en grupo, pero en nuestro caso, solo ha sido desarrollado por una persona. Por lo tanto, en ciertos momentos, se difiere en la metodología de desarrollo.

Las diferentes funcionalidades que encontramos en el proyecto han sido desarrolladas mediante iteraciones, pero estas no disponían de una duración determinada, es decir, podíamos encontrar tareas que duraban dos días o tareas que duraban una semana. Normalmente cada iteración se basaba en la planificación, análisis, diseño y codificación, defiriendo de la metodología en la creación de la documentación pues se dejó para el final del proyecto.

Para gestionar todas las tareas a realizar, hicimos uso de la herramienta Trello.

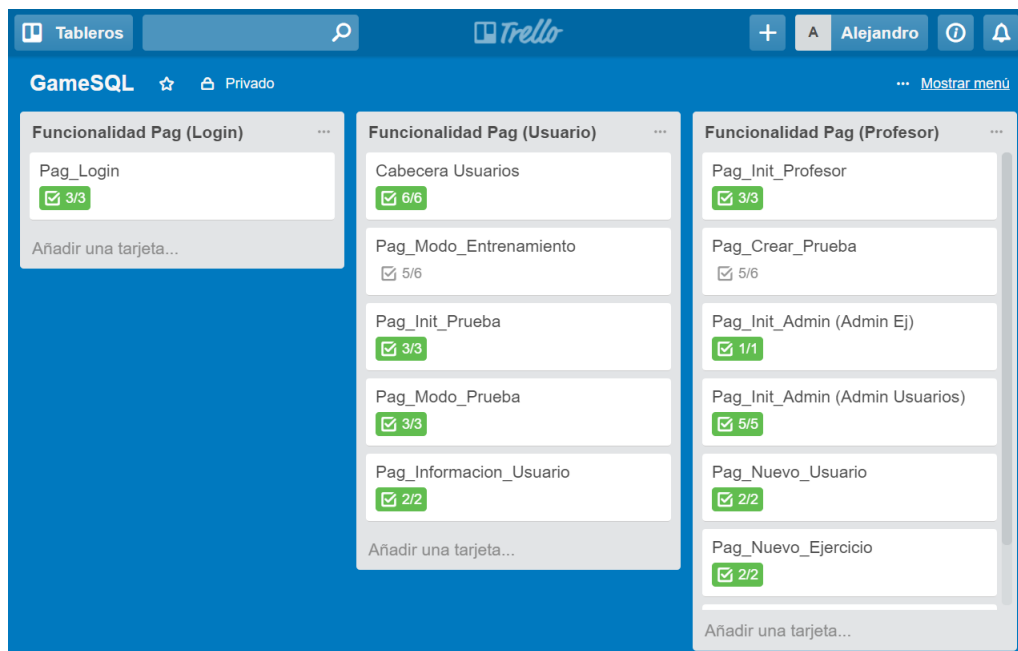


Figura 3: Tablón GameSQL en Trello.

Trello [9] es un aplicación web que nos permite organizar tareas usando una serie de tableros donde se publican las tareas a realizar. En nuestro caso, como vemos en la figura 3, teníamos un tablón para la aplicación GameSQL donde podíamos encontrar diferentes listas con una serie de tarjetas, donde cada tarjeta estaba formada por una lista de tareas.

La organización esta formada por:

Tablero GameSQL

- └ Listas..... Una por cada rol, más la vista de inicio de sesión
 - └ Tarjetas..... Una por cada una de las vistas
 - └ Listas de tareas..... Diferentes tareas que forman la vista

Una vez que las tareas eran realizadas, se comprobaban mediante su ejecución y si cumplía su funcionamiento se marcaba como realizada en su correspondiente apartado. En caso de que fallara, se corregía mediante técnicas de depuración, etc.

5. Diseño e implementación

5.1. Lenguajes usados

5.1.1. Java

Java [10] es un lenguaje de propósito general, concurrente y orientado a objetos que fue diseñado para que una vez que el desarrollador desarrolle su aplicación, no tenga que volver a escribir el programa si desea ejecutar su aplicación en otra plataforma, es decir, un código que se ejecuta en una plataforma no tiene que volver a ser compilado para ejecutarlo en otra.

A pesar de que hay muchos más lenguajes en el mercado, seleccionamos este pues a día de hoy pertenece a la misma compañía con la cual desarrollaremos la base de datos, logrando una total integración. Además de todas estas cualidades, actualmente, es el más popular para el uso de aplicaciones clientes - servidor, poseyendo una amplia API.

Entre sus cualidades cabe destacar:

- **Orientado a objetos**

Método de programación y diseño del lenguaje que permite diseñar el software de manera que los datos, estén unidos a sus operaciones, haciendo que un gran proyecto sea más fácil de gestionar y manejar, mejorando su calidad y permitiendo la reutilización del software.

- **Independencia de la plataforma**

El lenguaje Java puede ejecutarse en cualquier tipo de hardware y todo esto es gracias a que cuando compilamos el código, se genera un código llamado Java bytecode (un código que se encuentra entre código fuente y código máquina) y es ejecutado en una máquina virtual (JVM). Por lo tanto todas las máquinas que posean JVM, podrán hacer uso de él, independientemente de la arquitectura.

- **Recolector de basura**

En comparación con otros lenguajes, en Java el problema de las fugas de memoria se evita gracias a este sistema. Nosotros como programador podemos determinar cuando se crean los objetos, pero el entorno de tiempo de ejecución (Java runtime) se encarga de gestionar el ciclo de vida. Por lo tanto mientras un objeto esté referenciado se mantendrá en el ciclo y en cuanto pierda su referencia, el recolector de basura se encargará de eliminarlo de memoria.

5.1.2. SQL

Structure Query Language (SQL) [11], es un lenguaje de consultas estructurada que nos permite el acceso a un sistema de gestión de base de datos relacionales.

Este lenguaje fue diseñado exclusivamente para manejar datos que se encuentran almacenados en un sistema de bases de datos o para procesar flujos de datos en un sistema de gestión de flujos.

Para lograr realizar estas operaciones hace uso del álgebra y del cálculo relacional, haciendo posible que mediante una sentencia podemos realizar operaciones que en un lenguaje de registros equivaldría a uno o mas programas.

En el podemos encontrar:

- **Lenguaje de definición de datos (DDL)**

Es el que se encarga de la creación o modificación de la estructura que tendrá nuestra base de datos (tablas, relaciones, etc.), permitiendo crear, eliminar, modificar, etc.

- **Lenguaje de definición de datos (DML)**

Es el encargado de manipular los datos. Es decir, con el podemos consultar las tablas, insertar datos, eliminarlos, modificarlos, etc.

5.1.3. HTML

HyperText Markup Language (HTML) [12] es el encargado de la creación de la página web. En él se define la estructura que tendrá las diferentes vistas de nuestra aplicación web (texto, imágenes, tablas, vídeos, etc). Es un estándar a cargo del Consorcio WWW [13] y actualmente se considera el lenguaje más importante para la visualización de páginas webs.

5.1.4. CSS

Cascading Stylesheets (CSS) [14] es un lenguaje de diseño gráfico que se encarga de definir y crear la presentación de un documento estructurado como puede ser HTML, XHTML, etc. Estas hojas de estilos son un estándar que también están a cargo del Consorcio WWW y está principalmente diseñado para separar el contenido del documento y la forma de presentación.

A la hora de usar estas hojas de estilos, podemos hacerlo de tres maneras diferentes:

- **Externa:** Se crea una hoja de estilo (.css) y en la página HTML se llama mediante un link en su cabecera.

- Interna: En la propia página HTML se define una sección “style” en la cabecera.
- En línea: Se define el estilo en un elemento concreto usando el atributo “style”.

5.1.5. JavaScript

JavaScript [15] es un lenguaje de programación interpretado, definido como orientado a objetos, basados en prototipos, imperativo, débilmente tipado y dinámico. Se carga en el lado del cliente, por lo tanto el navegador tiene que ser capaz de soportarlo. Con el podemos mejorar la interfaz de usuarios y crear páginas webs dinámicas.

5.1.6. JQuery

JQuery [16] es una biblioteca multiplataforma de JavaScript que permite simplificar bastante la manera de interactuar con los documentos HTML. Por ejemplo nos permite manipular el árbol DOM, manejar eventos, crear animaciones e integrar la técnica AJAX.

5.2. Tecnologías utilizadas

5.2.1. JSP

JavaServer Pages [17] es una tecnología que nos ayuda a crear páginas webs dinámicas basadas en HTML. Esta nos permite combinar el código HTML con el código Java haciendo más fácil su usabilidad así como su mantenimiento, ya que por un lado tenemos el contenido estático de la plantilla HTML y por el otro el contenido dinámico.

La características principales que podemos encontrar son:

- Su ejecución se basa en una máquina virtual Java, haciendo posible una ejecución multiplataforma.
- Para su despliegue es necesario el uso de servidores webs que sean compatibles con contenedores servlet.
- Cada JSP se ejecuta en un hilo propio (contexto). Pero no es ejecutado hasta que recibe una petición, sino que las peticiones persisten haciendo que no se pierda el tiempo en invocaciones.
- Gracias a que posee persistencia permite crear conexiones a bases de datos y manejo de sesiones de forma eficiente.

En la estructura de una página JSP pueden aparecer 5 tipos de elementos:

- Elementos de scripting: permite añadir código Java mediante scriptlets, expresiones o declaraciones.

- Directivas: especifican información de la página mediante las etiquetas page, include o taglib.
- Expresiones EL
- Acciones: etiquetas para realizar algunas funciones.
- Código de plantilla: definición de la estructura estática de la página.

5.2.2. Apache Tomcat

Tomcat [18] es un contenedor web con soporte de servlets y JSPs que mediante el compilador que dispone (Jasper) es capaz de convertir los JSPs en servlets y normalmente van unidos con el servidor web Apache. Fue escrito en Java, así que puede funcionar en cualquier máquina que disponga la máquina virtual de Java.

5.2.3. Oracle Database

Oracle Database [19] es un sistema de gestión de base de datos de tipo objeto-relacional creado por Oracle en el cual nosotros desplegamos nuestra base de datos.

Para la creación del trabajo, usaremos una edición más ligera (Oracle Database Express Edition) que nos permite realizar todo lo necesario para su creación y puesta a punto. Esta versión funciona con funciones y características limitadas, pero para la creación del trabajo es más que suficiente. Una vez que el trabajo este desarrollado se tiene pensado desplegar en la versión completa.

5.3. Entornos de desarrollo utilizados

5.3.1. NetBeans IDE

A pesar de que existen muchos entornos de programación e incluso algunos más especializados que otros en algún tipo de lenguaje, hemos seleccionado NetBeans IDE porque nos permite realizar todas las tareas necesarias desde el.

NetBeans IDE [20] es un entorno de desarrollo de código abierto y lo más importante es que es gratuito. Con el podemos realizar todo lo relacionado con código Java, las páginas HTML, CSS, JavaScript e incluso manejar servicios como el servidor web Apache o la base de datos de una forma fácil y de manera integrada.

5.3.2. DataModeler

Oracle SQL Developer Data Modeler [21] es un entorno gráfico que nos permite crear o modificar el esquema que usaremos en nuestra base de datos. Es bastante intuitivo, fácil de usar y además al pertenecer a la misma compañía que nuestra base de datos, obtenemos una total integración. Con el podemos desarrollar el modelo lógico y relacional de nuestra base de datos, para luego exportarlo.

5.3.3. SQLDeveloper

Oracle SQL Developer [22] es un entorno de desarrollo que nos permite simplificar y facilitar todo el desarrollo y gestión de las bases de datos que necesitaremos para nuestro proyecto. En el podemos ejecutar todas nuestras consultas, crear script, importar desde Data Modeler, e infinidad de cosas más. Hemos seleccionado este entorno porque se integra completamente con Oracle y además es gratuito.

5.4. Framework utilizados

5.4.1. Primefaces

Primefaces [23] es un framework creado para JavaEE que nos permitirá darle un aspecto más visual a nuestra aplicación web. Es bastante fácil de usar y además dispone de un manual con ejemplos de uso, así como una gran comunidad que hace que se mantenga al día. En el podemos encontrar infinidad de componentes como tablas, botones, tipos de mensajes, etc y su integración es bastante fácil. Además posee temas ya creados por diferentes usuarios y permite una integración con móviles, tablets, etc.

5.4.2. FontAwesome y Google Fonts

Font Awesome [24] es un conjunto de herramientas que nos permite insertar infinidad de iconos en nuestra aplicación y diferentes tipos de fuentes, aunque para darle estilos a nuestras fuentes hemos usado Google Fonts [25], ya que posee un gran repertorio con ejemplos y diferentes posibilidades.

6. Funcionalidad requerida

6.1. Actores

6.1.1. Alumno

El alumno es el usuario principal de la aplicación web. Tendrá un perfil de jugador y podrá responder a las diferentes sentencias que se le proporcionarán, informar mediante un mensaje de error al profesor, responder las pruebas que el profesor cree, crear ejercicios para retos y ver diferente información sobre su perfil, así como los resultados de las pruebas que ha respondido.

6.1.2. Profesor

Es el usuario que se encarga de gestionar todo lo relevante con la asignatura. Podrá administrar los usuarios o los ejercicios, solucionar los errores que los alumnos han informado, crear retos basándose en los ejercicios que han sido creados por los alumnos como reto, crear pruebas para evaluar a los alumnos y poder ver los resultados, así como modificar estos resultados si es conveniente.

6.1.3. Administrador

El administrador es el encargado de gestionar la aplicación completa. Podrá crear profesores e indicar que asignaturas imparte, crear asignaturas y ejecutar procesos que se encuentran almacenados en la base de datos.

6.2. Requisitos funcionales

6.2.1. Requisitos funcionales comunes

RF1 Iniciar sesión

Tanto el alumno, el profesor como el administrador tienen que poder iniciar sesión en el sistema para poder hacer uso de las herramientas que este ofrece. Cada actor será redirigido a la parte de la aplicación que le corresponde, con sus diferentes funcionalidades.

Prerrequisito: Deben de estar dados de alta en el sistema.

RF2 Cerrar sesión

Todos los actores deben poder cerrar la sesión una vez que se ha iniciado esta.

Prerrequisito: El usuario tenga la sesión iniciada (RF1).

6.2.2. Requisitos funcionales para el actor “profesor”

RF3 Crear un usuario

Podrá crear un usuario, donde tendrá que introducir todos los datos referentes a la información del alumno y la información de matriculación.

Prerrequisito: Tener la sesión iniciada (RF1).

RF4 Crear un ejercicio

Podrá crear un ejercicio tanto para evaluación, reto o entrenamiento. Para ello deberá rellenar todos los campos obligatorios, así como añadir al menos una solución.

Prerrequisito: Tener la sesión iniciada (RF1).

RF5 Administración de usuarios

Tendrá la posibilidad de modificar toda la información referente a los diferentes alumnos disponibles matriculados en su asignatura, es decir, permite la modificación de la información del perfil, el borrado del avatar o volver a poner la contraseña definida como la contraseña por defecto, de cada alumno.

Prerrequisito: Los usuarios tienen que estar previamente creados (RF3).

RF6 Administración de ejercicios

Tendrá la posibilidad de modificar todos los campos relacionados con los ejercicios, añadir nuevas soluciones a este o idiomas. Además, dispone de la posibilidad de reiniciar los fallos e intentos del ejercicio seleccionado.

Prerrequisito: Los ejercicios tienen que estar previamente creado (RF4).

RF7 Validar reto

Podrá validar los ejercicios que han sido mandados por los usuarios en el modo reto, dándoles el estado de validados o rechazados. En caso de validados se deberá puntuar tanto al creador del reto, como a los posibles usuarios que intenten solucionarlos.

Prerrequisito: El usuario ha debido crear un ejercicio de reto previamente (RF18).

RF8 Crear un reto

Tendrá la posibilidad de crear retos, basándose en los ejercicios que han sido mandados por el usuario y validados como aceptados. Para ello tendrá que rellenar los campos necesarios y seleccionar el conjunto de ejercicios que lo compondrá.

Prerrequisito: El usuario ha debido crear al menos un ejercicio de reto previamente (RF18) estar con estado validado (RF7).

RF9 Crear una prueba evaluable

Tendrá la posibilidad de crear pruebas evaluables para sus alumnos, las cuales estarán disponibles durante un periodo de tiempo.

Prerrequisito: Se ha debido crear ejercicios de modo prueba previamente (RF3).

RF10 Consultar informe de errores

Podrá consultar los informes que han sido mandado por sus alumnos referente a los ejercicios. Estos ejercicios pueden ser modificados directamente.

Prerrequisito: El usuario previamente debe informar sobre un error (RF17) en un ejercicio previamente creado (RF4).

RF11 Consultar las prueba de los alumnos

Podrá consultar en todo momento las soluciones que han aportado los alumnos a los ejercicios de las pruebas evaluables, así como la modificación de la puntuación si es necesario.

Prerrequisito: La prueba ha tenido que ser previamente creada (RF9).

RF12 Corregir pruebas

Podrá mandar al sistema la orden de que corrija una prueba previamente seleccionada.

Prerrequisito: La prueba ha tenido que ser previamente creada (RF9).

RF13 Habilitar/Deshabilitar ver pruebas

Podrá mandar la orden al sistema de que los resultados de la corrección, estén o no disponibles para los usuarios.

Prerrequisito: La prueba debe estar previamente creada (RF9) y corregida (RF12).

RF14 Subir alumnos y ejercicios mediante archivos .csv

Podrá dar de alta en el sistema a alumnos y crear ejercicios mediante un archivo CSV.

6.2.3. Requisitos funcionales para el actor “alumno”

RF15 Responder ejercicios de entrenamiento

Podrá responder a los ejercicios mediante la inserción de sentencias SQL, las cuales no pueden contener palabras reservadas. Obtendrá la puntuación de manera directa.

Prerrequisito: El ejercicio debe estar creado previamente por el profesor (RF4).

RF16 Responder ejercicios de evaluación

Podrá responder a cada ejercicio que componen la prueba evaluable mediante la inserción de sentencia SQL, con la prohibición de insertar sentencia con palabras reservadas. El alumno no obtendrá la puntuación directamente.

Prerrequisito: El ejercicio debe estar creado previamente por el profesor (RF4) y la prueba evaluable (RF9).

RF17 Informar de un error en un ejercicio

Tendrá a su disposición un apartado por cada ejercicio en el modo entrenamiento donde podrá en cualquier momento informar sobre un error encontrado en dicho ejercicio.

Prerrequisito: El ejercicio debe estar creado previamente por el profesor (RF4).

RF18 Crear consultas para retos

Tendrá la posibilidad de crear consultas que luego pueden ser enviadas por el profesor como retos.

Prerrequisito: Tener la sesión iniciada (RF1).

RF19 Consultar información de las pruebas

Podrá consultar los resultados obtenidos en las pruebas realizadas.

Prerrequisito: La prueba ha tenido que ser corregida por el sistema previamente (RF12).

RF20 Consultar información del perfil

En todo momento el usuario podrá consultar la información sobre su perfil, así como la posible modificación de alguna.

Prerrequisito: El usuario debe estar creado (RF3) y la sesión iniciada (RF1).

RF21 Cambiar contraseña

En todo momento el usuario dispondrá de la posibilidad de cambiar la contraseña que le permite iniciar sesión en el sistema.

Prerrequisito: El usuario debe estar creado (RF3), la sesión iniciada (RF1) y es necesario la inserción de la contraseña antigua.

RF22 Cambiar avatar

En todo momento el usuario podrá modificar el avatar por otro e incluso la eliminación de este, dejando el avatar por defecto.

Prerrequisito: El usuario debe estar creado (RF3) y la sesión iniciada (RF1).

6.2.4. Requisitos funcionales para el actor “administrador”

RF23 Crear un profesor

Podrá crear un profesor, al cual se le dará el alta en el sistema.

Prerrequisito: El profesor no debe existir con anterioridad.

RF24 Borrar un profesor

Podrá borrar un profesor, al cual se le dará de baja en el sistema.

Prerrequisito: El profesor debe haber sido creado previamente (RF23).

RF25 Modificar un profesor

En todo momento podrá editar la información relevante a un profesor e indicar que asignaturas impartirá.

Prerrequisito: El profesor debe haber sido creado previamente (RF23) y la asignatura (RF26).

RF26 Crear un asignatura

Podrá crear una asignatura que luego podrá ser asignada a un profesor.

Prerrequisito: La asignatura no debe existir con anterioridad.

RF27 Borrar asignatura

Podrá borrar una asignatura.

Prerrequisito: Las asignatura debe estar creada (RF26), no debe ser impartida por ningún profesor (RF25) ni que ningún alumno este matriculado en ella (RF3).

RF28 Modificar asignatura

Podrá modificar la información relevante a la asignatura.

Prerrequisito: La asignatura debe estar previamente creada (RF26).

RF29 Crear un proceso

Podrá crear un proceso, que se ejecutará sobre la base de datos indicada.

Prerrequisito: El proceso debe encontrarse almacenado en la base de datos sobre la que se ejecuta.

RF30 Borrar proceso

Podrá borrar un proceso.

Prerrequisito: El proceso debe estar previamente creado (RF29).

RF31 Ejecutar un proceso

Podrá ejecutar un proceso con los argumentos oportunos.

Prerrequisito: El proceso debe estar previamente creado (RF29).

6.3. Requisitos no funcionales

RNF1 Disponibilidad

Al ser una aplicación web se garantiza una total disponibilidad en diferentes dispositivos siempre que se disponga de una conexión a internet, se tenga acceso a la red donde se despliega y un navegador web.

RNF2 Accesibilidad simultánea

El sistema permite una accesibilidad simultánea, para garantizar un sistema multiusuario.

RNF3 Robustez

El sistema está capacitado para responder ante situaciones imprevistas, como la entrada indebida de datos, informando con diferentes tipos de mensajes.

RNF4 Extensibilidad

El sistema es fácilmente extensible, pudiendo añadir más funciones, tanto nuevas o mejorarlas sin tener que afectar a las ya creadas.

RNF5 Usabilidad

El sistema es fácilmente usable desde el punto de vista del usuario, disponiendo de una gran amplia ayuda en la misma aplicación. Lo que se busca es un fácil manejo y un uso con un fácil aprendizaje, para no interferir en el objetivo final de esta aplicación web, que es el aprendizaje del lenguaje SQL.

RNF6 Interfaz

La interfaz visible para los usuarios es visualmente agradable e intuitiva disponiendo de un mensaje que explica la acción a realizar en los botones. Por otra parte, la interfaz para el profesor/administrador es visualmente más simple, pues nos centramos más en la usabilidad.

RNF7 Seguridad

El sistema dispone de un control de inicio de sesión mediante un usuario y contraseña el cual impide que otros usuarios puedan acceder a su cuenta, pudiendo interferir en algunas acciones. El control se lleva a cabo mediante la conexión con la base de datos, por lo tanto, el usuario debe tener todos los permisos necesarios que están disponibles en el rol usuario. Para garantizar más seguridad se dispone de un control que impide el acceso mediante la copia de urls.

7. Diseño de la aplicación

7.1. Casos de uso

Basándonos en los requisitos funcionales descritos en el apartado 6.2 podemos obtener los casos de usos del Lenguaje de Modelado Unificado (UML) [26], los cuales normalmente se representan mediante diagramas que nos indican cómo el sistema se comporta a la hora de afrontar estos requisitos, pero sin llegar a conocer su implementación.

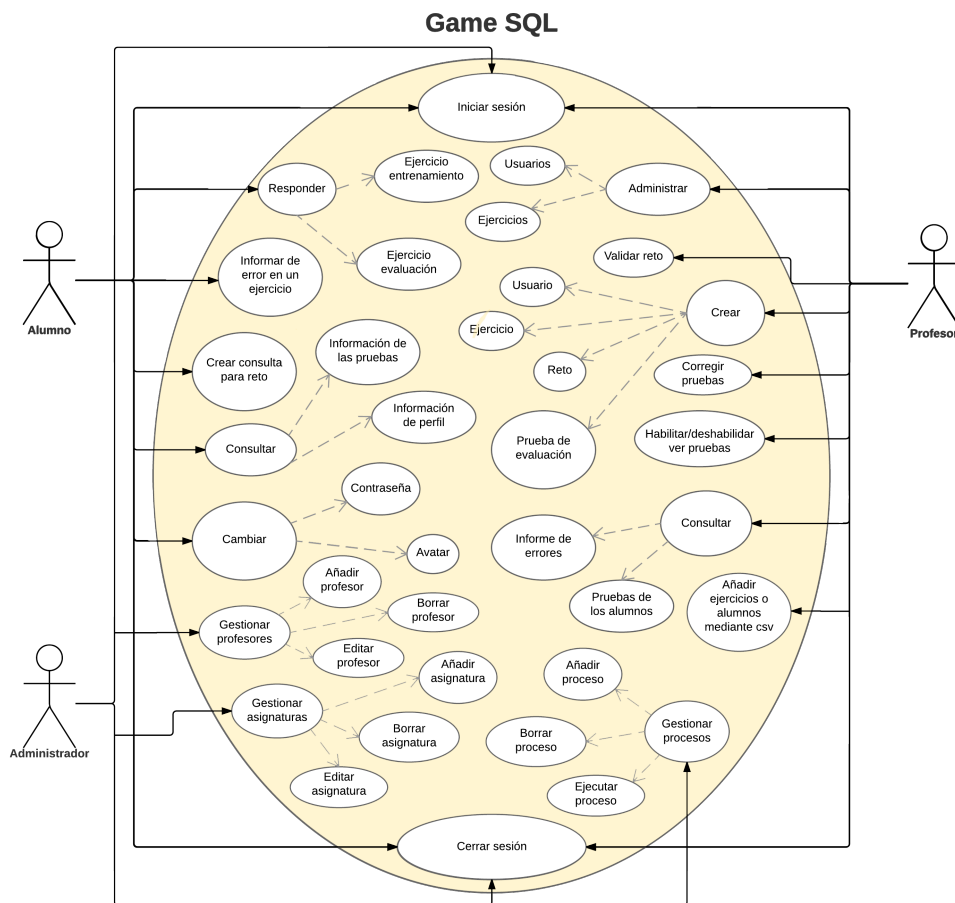


Figura 4: Casos de uso general.

En la figura 4 podemos ver el diagrama de casos de uso genérico de nuestra aplicación web, pero en la figuras 5, 6 y 7 podemos ver de una manera más detallada los casos de uso para el alumno, profesor y administrador, respectivamente.

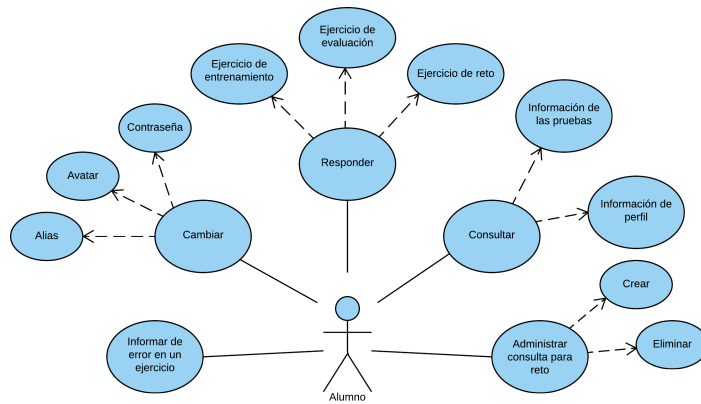


Figura 5: Casos de uso del actor alumno.

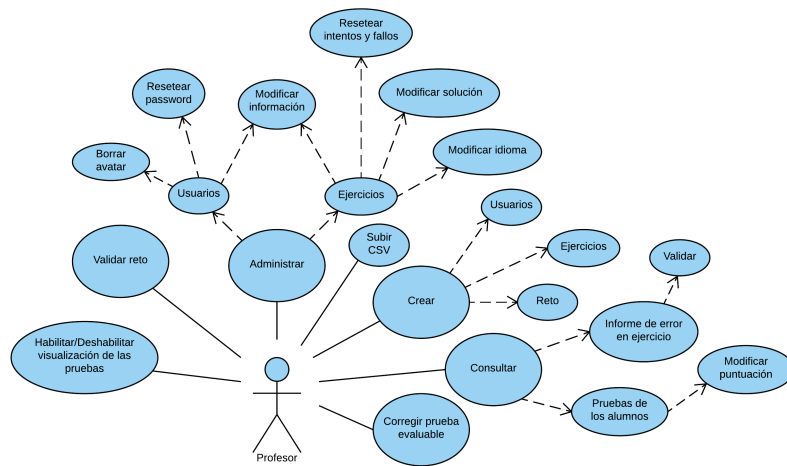


Figura 6: Casos de uso del actor profesor.

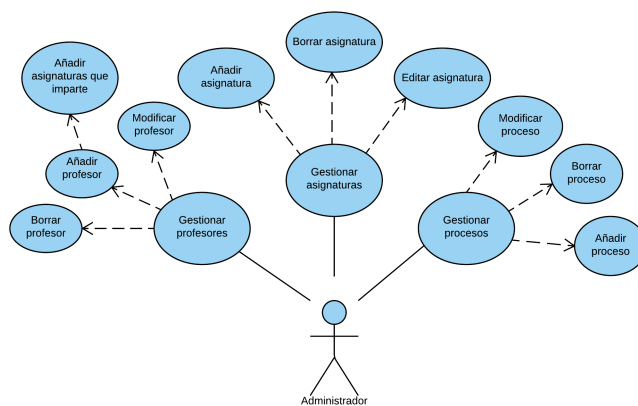


Figura 7: Casos de uso del actor administrador.

7.2. Arquitectura

La arquitectura es uno de los puntos más fuertes a la hora de diseñar la aplicación, pues es lo que determina el funcionamiento y hace posible que se cubran todas las necesidades.

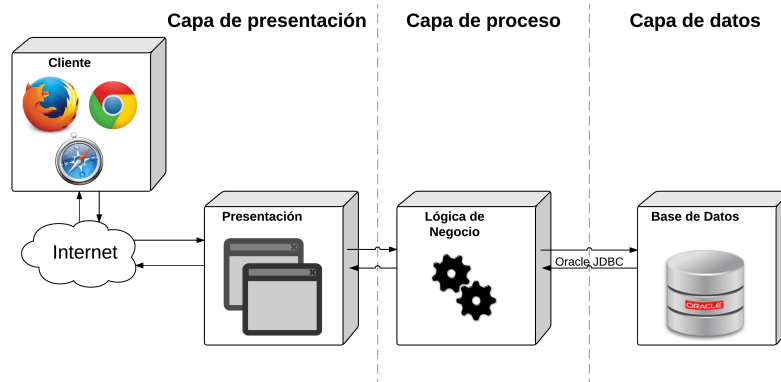


Figura 8: Modelo en tres capas.

A parte de que el diseño cumpla con todas las restricciones y necesidades que busquemos, también es conveniente seleccionar una arquitectura que sea fácil de mantener, flexible y fácil de integrar. Nosotros al estar desarrollando una aplicación web, hemos optado por una arquitectura web.

Este modelo se basa en el modelo de tres capas, el cual permite que los datos sean manejados de forma separada a la interfaz del usuario. Esto es posible gracias a la capa intermedia que llamamos capa de negocio.

Las 3 capas mencionadas son:

- **Capa de presentación**

En esta capa podemos encontrar el lado del cliente, el cual se encarga de recoger la información del usuario y enviarlo posteriormente al servidor, así como visualizar el contenido. Por el otro lado, tenemos el lado del servidor el cual se encarga de generar la parte visual con el procesamiento de los datos.

- **Capa de proceso**

Se encarga de recibir los datos de la capa de presentación, procesarlos con las operaciones oportunas y mandarlos a la capa de presentación o a la capa de datos.

- **Capa de datos**

Se encarga de almacenar los datos que han sido recogidos de la capa de proceso para garantizar su integridad y futuro uso.

7.3. Estructura de la base de datos

En la capa de datos nosotros hemos optado por el uso de la base de datos Oracle, pues es la que se usa en la Universidad de Málaga (UMA) a día de hoy para la impartición de las asignaturas donde se aprende el lenguaje SQL.

La conexión entre la capa de proceso y la capa de datos, se hace mediante Oracle JDBC Drivers. JDBC (*Java Database Connectivity*) [27] es una API que se encarga de ejecutar operaciones sobre la base de datos desde el lenguaje de programación Java, independientemente del sistema operativo en el cual se esté ejecutando.

Para ello se aporta una serie de librerías las cuales te permite realizar todo tipo de tareas relacionadas con las bases de datos. Como consultar, borrar, crear, modificar tablas e incluso la ejecución de procedimientos almacenados, etc.

7.3.1. Modelo de datos conceptual

Este modelo, también llamado modelo lógico nos ofrece una descripción de la estructura de datos que usaremos en nuestra base de datos, así como las restricciones de integridad. En la figura 9 se muestra esta estructura con un modelo entidad-relación.

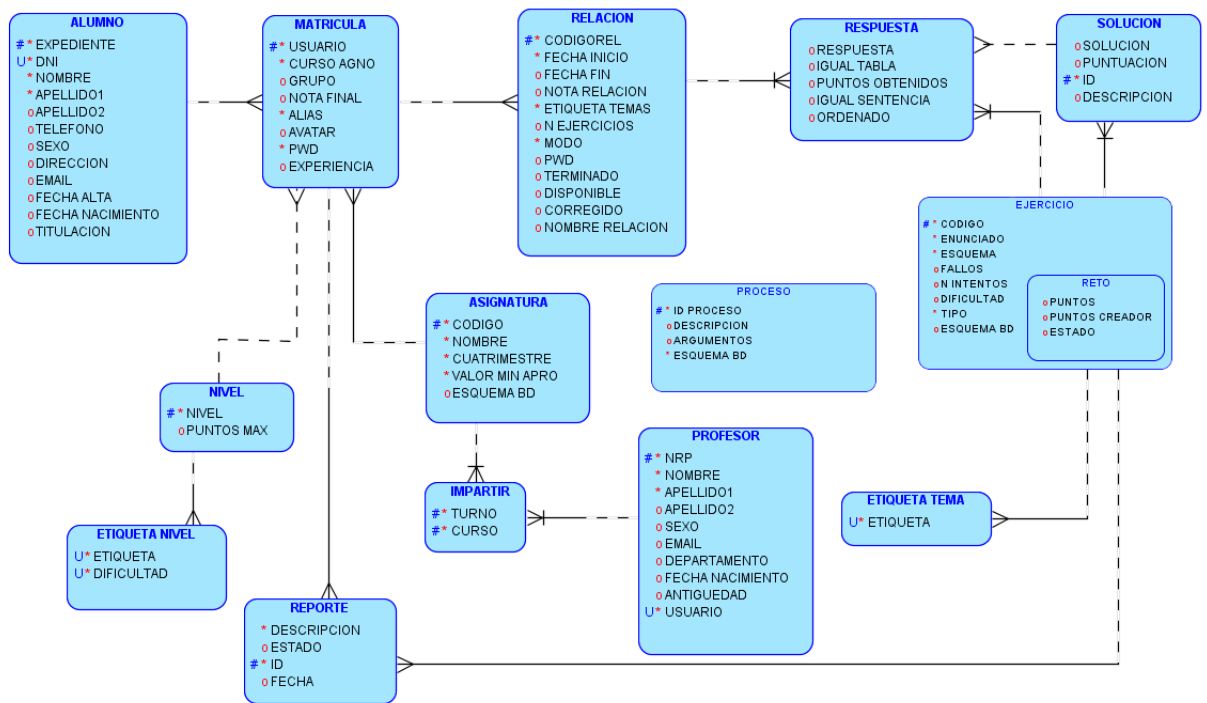


Figura 9: Modelo de datos conceptual.

7.3.2. Modelo relacional

A partir del modelo de datos conceptual, podemos obtener el modelo relacional el cual nos permitirá modelar y gestionar la base de datos. En la figura 10 se muestra su estructura.

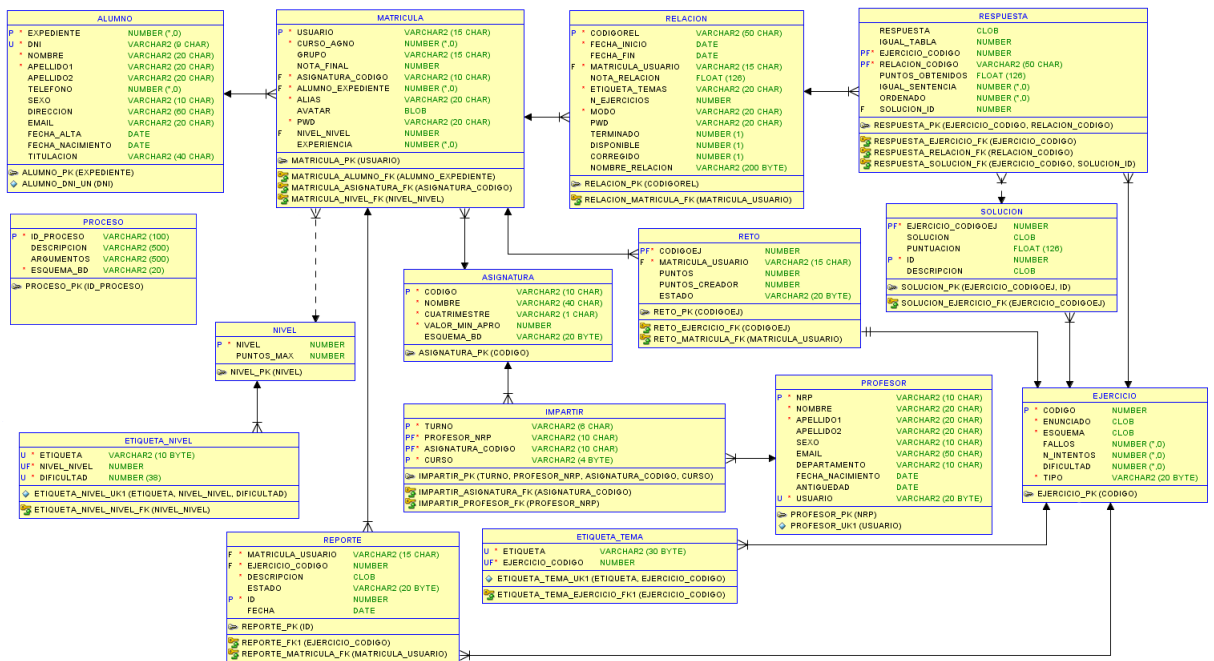


Figura 10: Modelo relacional.

7.3.3. Modelo o modelos externos

La aplicación web se ha diseñado para ejecutar las consultas sql en diferentes modelos, es decir, además del modelo que hace que el funcionamiento de la aplicación funcione (secciones 7.3.1 y 7.3.2), es necesario uno o más modelos, sobre los cuales se obtendrá las tablas y datos tanto para las respuestas proporcionadas por el alumno, como las tablas y datos de las respuestas guardadas en la base de datos.

7.4. Estructura del proyecto

Para garantizar un orden y mayor facilidad a la hora del uso del código, la aplicación ha sido descompuesta en diferentes directorios y clases. Al tener una gran cantidad de clases y funciones, solo nos centraremos en las más significativas para la aplicación.

Por un lado podemos encontrar los directorios que están directamente relacionados con la aplicación web y por el otro los procedimientos que nos permitirán realizar ciertas operaciones en nuestra base de datos, es decir, procedimientos necesarios para la funcionalidad de nuestra aplicación.

El lado de la aplicación web consta de la raíz principal, nuestro proyecto denominado GameSQL, del cual salen diferentes directorios:

- **Web Page**

Contiene las diferentes vistas que podemos encontrar en nuestra aplicación en un formato “.xhtml”, así como los directorios css (proporciona estilo a nuestras vistas), template (donde encontramos las cabecera de nuestras vistas), resources (imágenes estáticas, librería Font Awesome y una carpeta temporal).

- **Source Package**

Encontramos todo lo necesario para aportar la funcionalidad a nuestra aplicación. Entre los diferentes paquetes encontramos el paquete “Beans” que se encarga de la parte lógica de cada vista, el paquete “Entidad” que nos permite modelar el sistema y el paquete “Utilidades” que nos proporciona diferentes clases.

Y por el otro lado tenemos los paquetes que proceden de los procedimientos que tenemos almacenados en nuestra base de datos, que nos proporcionan ciertas funcionalidades. Entre estas funcionalidades encontramos la más importante que es el sistema de corrección. Esta funcionalidad será descrita en el apartado 7.5.

7.5. Sistema de corrección

Uno de los puntos más importante de nuestra aplicación web es el sistema de corrección, ya que hace posible que el sistema funcione. Sin el sistema de corrección la aplicación no tendría sentido, ya que el objetivo principal es que el alumno reciba el estímulo, en nuestro caso una puntuación, en el menor tiempo posible para así

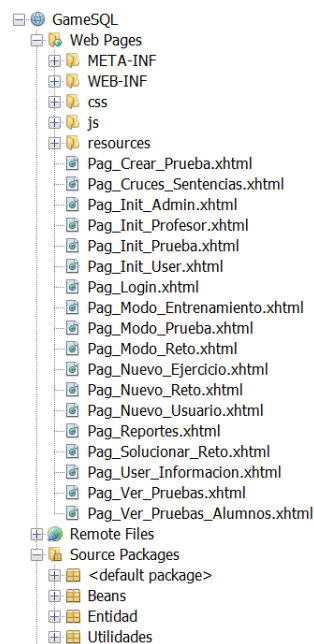


Figura 11: Directorios.

lograr un efecto acción-reacción.

Para lograr nuestro objetivo hemos creado un sistema de corrección, basándonos en dos técnicas que describiremos a continuación en los apartados 7.5.1 y 7.5.2. En ambos casos necesitamos la sentencia que el alumno a proporcionado y la sentencia respuesta que se encuentra en nuestra base de datos. Estas sentencias no son más que cadenas de texto, por lo tanto, las podemos tratar como tal.

7.5.1. Misma sentencia

Esta técnica fue usada para ahorrar tiempo y procesamiento al sistema, ya que la técnica del apartado 7.5.2 necesita más recursos.

Su funcionamiento se basa en comparar las dos cadenas proporcionadas, es decir, compara la cadena proporcionada por el usuario con la cadena guardada en la base de datos. Para realizar esta comparación, la cadena proporcionada por el usuario es pasada por unos filtros donde se eliminan todos los posibles espacios y se realiza de manera insensible a mayúsculas y minúsculas.

Como el sistema permite guardar diferentes sentencias como respuesta correcta, se va iterando hasta encontrar la correcta. Si se da el caso de que es correcta, se guarda la puntuación que esté relacionada con esa respuesta y en el caso de que no existan más iteraciones se dará como incorrecta, evaluándola con un cero.

7.5.2. Comparación de tablas

La manera de guardar los datos en las bases de datos es mediante tablas y una de las principales características del lenguaje SQL es poder realizar operaciones algebraicas sobre ellas. Nosotros hemos aprovechado esta característica y la corrección la realizamos mediante operaciones de conjuntos.

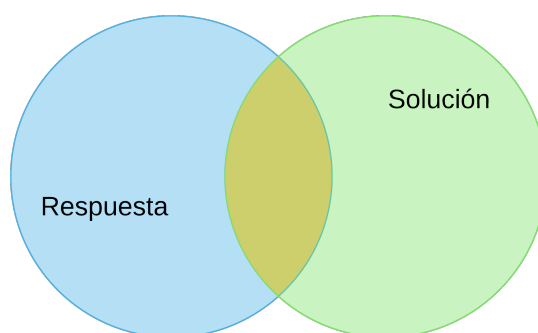


Figura 12: Diagrama de Venn.

A la hora de corregir tendremos la cadena “respuesta del alumno” y la cadena “solución” almacenada en la base de datos, las cuales son sentencias SQL. Estas sentencias son ejecutadas sobre un modelo externo (sección 7.3.3) y nos devolverá unas tablas sobre las cuales podremos realizar operaciones de conjuntos. A estas

tablas las llamaremos respuesta y solución.

La operación de corrección, se basa en que el conjunto formado por la diferencia entre la tabla solución y la tabla respuesta y la diferencia entre la tabla respuesta y la tabla solución sea vacío. En la figura 12 se representa la operación mediante un diagrama de Venn y usando el lenguaje matemático sería:

$$((\text{Tabla respuesta} - \text{Tabla solución}) \cup (\text{Tabla solución} - \text{Tabla respuesta})) = \emptyset$$

En el caso de que el resultado de la tabla tenga que ser ordenado, una vez que comprobamos que las tablas son las mismas, comparamos fila a fila entre la tabla respuesta y la tabla solución. Si una fila no se encuentra en el mismo orden, se termina la ejecución y se valida como incorrecto.

7.6. Seguridad

Un factor a tener en cuenta a la hora de diseñar una aplicación web es la seguridad, ya que no podemos dejar que cualquier usuario acceda a esta. Si no se logra una seguridad mínima, cualquier usuario podría interferir en los datos y hacer la aplicación inutilizable.

Como mencionamos en la sección 7.3.3, nuestra aplicación está compuesta por un modelo de base de datos que hace posible el funcionamiento de la aplicación y de uno o varios modelos sobre los cuales se obtienen las tablas y datos para la corrección.

7.6.1. Seguridad mediante permisos

Para poder gestionar la seguridad de la aplicación web tenemos que tener en cuenta que se deben seleccionar los permisos adecuados para la aplicación en sí, así como para cada actor que podemos encontrar (sección 6.1).

Por lo tanto, los permisos para cada uno son:

Alumno Permisos concedidos al rol “ROL_ALUMNO”.

Profesor Permisos concedidos al rol “ROL_PROFESOR”.

Administrador Permisos concedidos al rol “ROL_ADMINISTRADOR”.

Aplicación Permisos de SYS.

Entre los permisos concedidos a los roles “ROL_PROFESOR”, “ROL_ALUMNO” y “ROL_ADMINISTRADOR” podemos encontrar permisos para ver tablas del modelo o modelos externos, modificar, ver vistas, iniciar sesión, ejecutar procedimientos almacenados, etc. Mientras que los permisos que engloban a SYS, son casi todos los existentes.

7.6.2. Seguridad mediante el control de las entradas

Como mencionamos en la sección 7.5, el sistema de corrección se basa en ejecutar cada sentencia y realizar una comparación de tablas. Esto puede llegar a ser muy peligroso, pues el alumno podría introducir una sentencia donde le indicara al sistema que borre todas las tablas y al iniciar el sistema de corrección, se ejecutaría y borraría todo.

Para evitar este problema se ha diseñado un procedimiento por el cual cada sentencia que es introducida, primero pasa por una serie de filtros donde se analiza la cadena de texto en busca de palabras reservadas como *delete*, *drop*, etc.

En el caso de que el alumno inserte este tipo de palabras, se avisa con un mensaje y además no se introduce en la base de datos, para evitar futuros problemas.

7.6.3. Seguridad mediante el control de ejecución

La estructura de la base de datos, se basa en un modelo que define la aplicación y luego un o unos modelos externos (sección 7.3.3) sobre los cuales se realizan las operaciones de corrección.

Este sistema principalmente se realizó para que las sentencias que el alumno tiene que ejecutar no se tengan que realizar sobre el modelo de la aplicación, pues para realizar esto se tendría que tener permisos para ver dichas tablas y esto puede dar lugar a problemas de seguridad.

Estos modelos son externos a la aplicación, es decir, el alumno puede tener todos los permisos de lectura que le concedería el poder de ver los datos y tablas disponibles. El alumno debe tener permiso de lectura de estos esquemas externos, pues para realizar estas consultas es necesario que el alumno conozca el esquema sobre el cual trabaja.

Su funcionamiento es simple, pues solo es necesario indicar sobre que modelo se hará la corrección. Para eso en la base de datos, cada asignatura posee un campo que indica en que esquema debe ser ejecutado y cada modelo externo tiene que tener un procedimiento almacenado que corrige las sentencias.

Con este sistema el alumno solo podría ver el modelo de esquema de su asignatura y no interferir sobre el de la aplicación.

7.7. Interfaz

A la hora de desarrollar la interfaz hemos tenido en cuenta varios aspectos como la usabilidad y la amigabilidad, ya que como hemos mencionado anteriormente, lo que se pretende es garantizar el aprendizaje, y sin estas dos cualidades puede que el alumno vea su aprendizaje como una carga, dando lugar al abandono.

Como hemos mencionado en la sección 6.1, en nuestra aplicación existen los roles alumno, profesor y administrador, y cada uno dispone de diferentes funcionalidades.

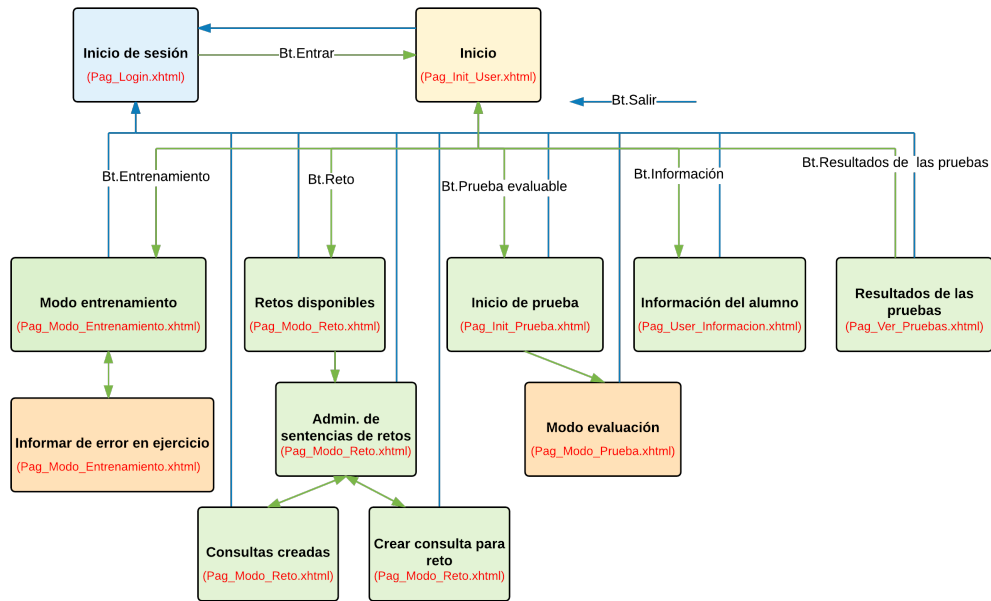


Figura 13: Flujo de la interfaz para el rol “alumno”.

El usuario principal de la aplicación es el alumno, por lo tanto, a la hora de desarrollar la interfaz para este rol, se hizo de manera más amena e intuitiva, mientras que las interfaces para el rol profesor y administrador son simple, ya que nos centramos en la funcionalidad.

En las figuras 13, 14 y 15 podemos ver el flujo que tiene la interfaz para el rol alumno y profesor respectivamente.

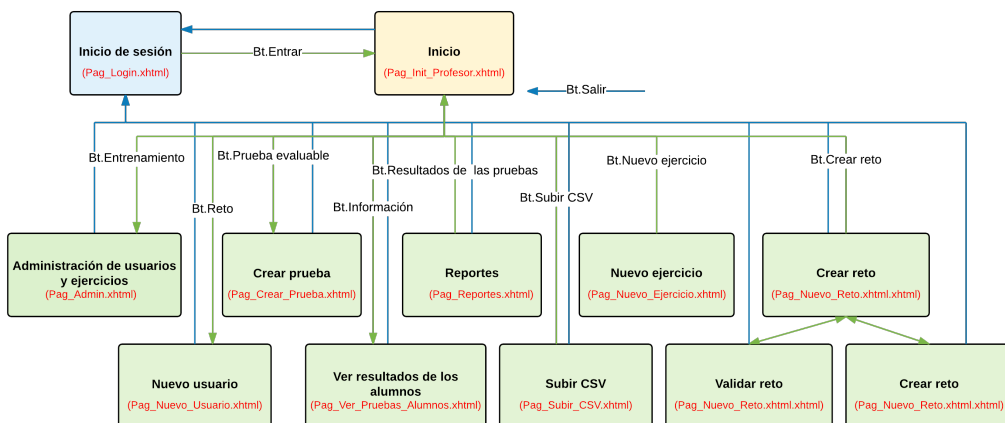


Figura 14: Flujo de la interfaz para el rol “profesor”.

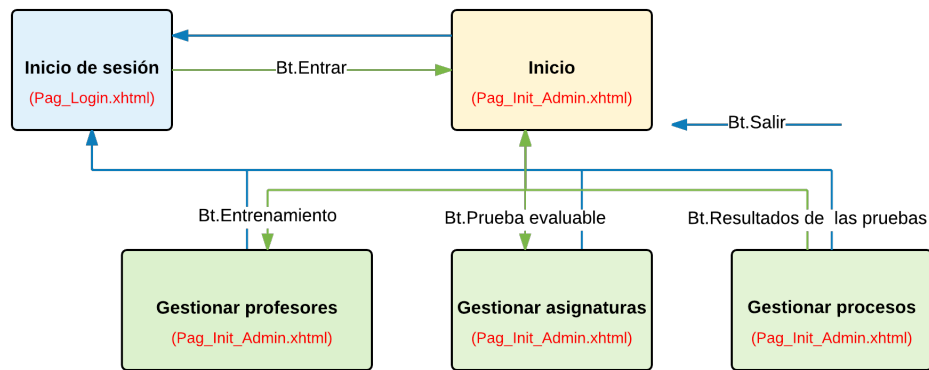


Figura 15: Flujo de la interfaz para el rol “administrador”.

A continuación, iremos viendo cada una de las vistas y definiremos lo más importante de cada una.

7.7.1. Inicio de sesión

Para poder ingresar en nuestra aplicación es necesario iniciar sesión en el sistema, y esto se logra mediante esta vista (figura 16).

En ella encontramos:

Campo 1 Formulario donde debemos introducir nuestro usuario y contraseña.

Campo 2 Botón con el cual podremos validar los datos introducidos en el campo 1. En caso incorrecto, se avisa con un mensaje de error.



Figura 16: Vista de inicio de sesión.

7.7.2. Interfaz del rol “alumno”

7.7.2.1 Cabecera de las vista del rol “alumno”

Para reutilizar código, definimos una cabecera (figura 17) que es común para todas las vistas pertenecientes al rol alumno.

En ella encontramos:

Campo 1 Botón de inicio que nos lleva a la vista inicial. Este botón está en todas las vistas visible menos en la vista inicio, ya que sería redundante.

Campo 2 Botón para cerrar la sesión.

Campo 3 Se muestra el nivel actual del usuario, así como la cantidad para alcanzar el siguiente.

Campo 4 Alias del usuario.



Figura 17: Cabecera para las vistas del rol “alumno”.

7.7.2.2 Vista de inicio

La vista inicial del rol alumno (figura 18), es la que nos permite navegar a las diferentes vistas de la aplicación y acceder a sus funcionalidades.

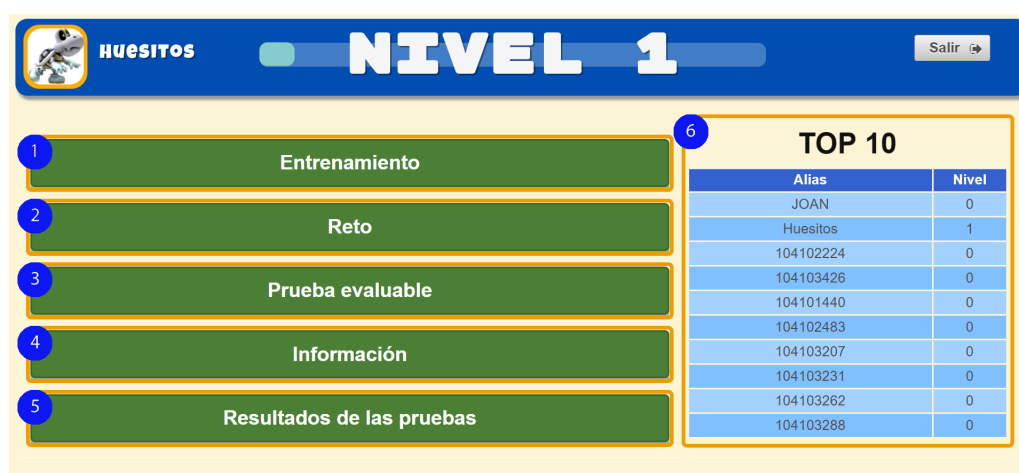


Figura 18: Vista de inicio del rol “alumno”.

En ella podemos ver:

Campo 1 Botón que nos permite acceder al modo entrenamiento.

Campo 2 Botón que nos permite acceder al modo reto.

Campo 3 Botón que nos permite acceder a las pruebas evaluables disponibles.

Campo 4 Botón que nos permite ver la información del perfil.

Campo 5 Botón que nos permite ver los resultados obtenidos de las pruebas hechas y corregidas.

Campo 6 Tabla “Top 10”, que muestra los usuario con más nivel y puntuación.

7.7.2.3 Vista del modo entrenamiento

El modo entrenamiento (figura 19) está formado por un conjunto de sentencias que serán mostradas para que el usuario pueda ir respondiéndolas. Una vez que ha respondido un porcentaje de ese conjunto podrá obtener la puntuación y pasar a otra relación.



Figura 19: Vista del modo entrenamiento.

Los campos que podemos ver son:

Campo 1 Información sobre la sentencia a responder.

Campo 2 Campo donde el usuario podrá responder con una sentencia sql.

Campo 3 Botón donde el usuario podrá informar al profesor sobre un error.

Campo 4 Botón para pasar a la pregunta anterior.

Campo 5 Botón que guardará la respuesta en la base de datos y además procederá a corregirla. Informará de la puntuación.

Campo 6 Botón para pasar a la pregunta siguiente.

Campo 7 Tabla donde se muestra el estado de cada pregunta.

Campo 8 Botón que se activará una vez que se ha superado el porcentaje de preguntas respondidas. Te permite terminar esa relación, obtener la puntuación y empezar otra.

7.7.2.4 Vista al subir de nivel y aumentar puntuación

Cada vez que se termina un conjunto de sentencias, se validan y se recibe una puntuación que se obtiene de la suma de todas las puntuaciones obtenidas por cada sentencia, al igual que se indica cuando se sube de nivel (figura 20).

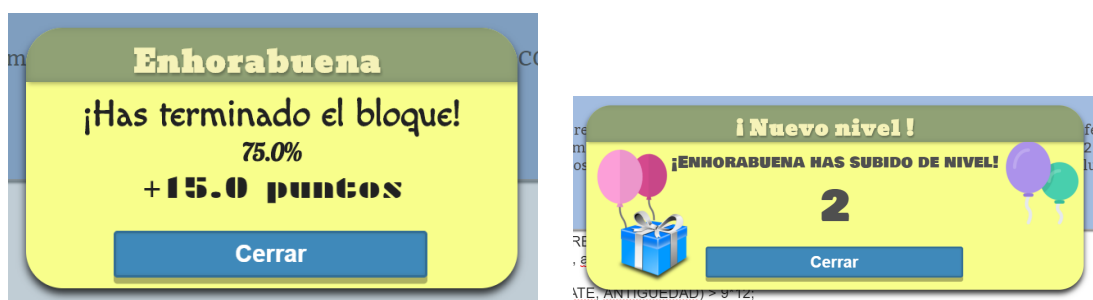


Figura 20: Vistas al subir de nivel y aumentar puntuación.

7.7.2.5 Vista para informar de un error en un ejercicio

Esta vista (figura 21) permite al usuario informar sobre un error en un ejercicio, este error luego puede ser visto por el profesor y corregido.

Esta formado por:

Campo 1 Formulario donde el usuario debe insertar un mensaje que describa el error y los botones para enviar o cancelar.

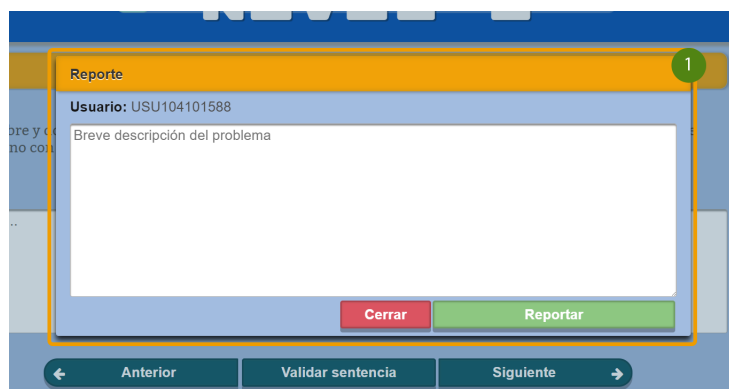


Figura 21: Vista para informar de error en un ejercicio.

7.7.2.6 Vista de los retos disponibles

En esta vista (figura 22) podemos ver los retos que otros alumnos han enviado y han sido aceptados por los alumnos. Estos retos se encontrarán a la espera de ser realizador por el alumno y una vez concluidos permite obtener una puntuación.

Podemos encontrar:

Campo 1 Botón que nos permite ir a la vista para administrar las consultas para retos.

Campo 2 Listado de las pruebas disponibles que el alumno puede realizar.

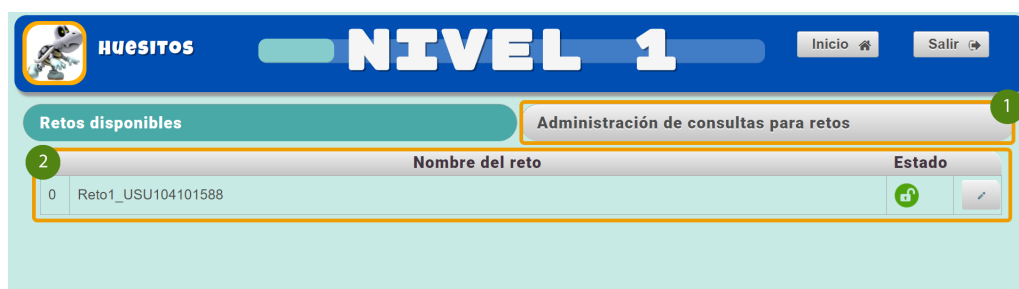


Figura 22: Vista de los retos disponibles.

7.7.2.7 Vista de las consultas creadas

Esta vista (figura 23) nos permite administrar los retos que hemos creado.

Podemos encontrar:

Campo 1 Listado de las consultas que se han enviado como posible reto. Estas consultas tienen que ser validadas por el profesor para que sean aceptadas o denegadas. Mientras se encuentran en estado de espera de validación, pueden ser eliminadas.

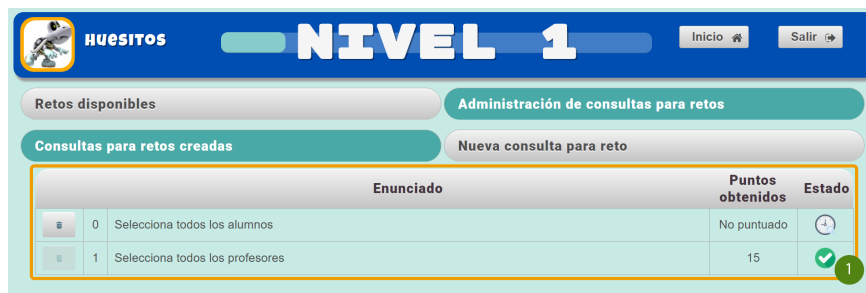


Figura 23: Vista de las consultas creadas.

7.7.2.8 Vista para crear una consulta de reto

Esta vista (figura 24) permite al alumno mandar un reto que una vez validado por el profesor, otros alumnos podrán contestar.

Campo 1 Formulario donde se debe introducir la información necesaria para crear el reto.

Campo 2 Botón para mandar el reto a la espera de validación.

The screenshot shows a web interface for 'NIVEL 1' under the 'HUESITOS' brand. At the top, there are navigation links for 'Inicio' and 'Salir'. Below the header, there are four buttons: 'Retos disponibles', 'Administración de consultas para retos', 'Consultas para retos creadas', and 'Nueva consulta para reto'. The main content area is divided into two columns for 'Enunciado:' and 'Solución:', with a 'Esquema:' field below. A 'Mandar reto' button is located at the bottom. Green circles with numbers 1 and 2 highlight the form fields and the button respectively.

Figura 24: Vista para crear un consulta de reto.

7.7.2.9 Vista al iniciar una prueba

Para poder realizar una prueba es necesario introducir una palabra de paso, así como seleccionar la prueba a realizar.

The screenshot shows a web interface for 'NIVEL 1' under the 'HUESITOS' brand. At the top, there are navigation links for 'Inicio' and 'Salir'. Below the header, there is a 'Normas' section with two warning icons and the following text: 'Esta prohibido el uso de cualquier tipo de dispositivo electrónico para consultar. No esta permitido hablar con el compañero. Una vez terminado cada sentencia, debe darle a guardar. Finalmente darle a terminar.' Below the norms section, there is a dropdown menu for 'Selecciona una prueba:', a password field for 'Palabra de paso:', and an 'Entrar' button. Green circles with numbers 1, 2, and 3 highlight the norms section, the dropdown menu, and the 'Entrar' button respectively.

Figura 25: Vista al iniciar una prueba.

En esta vista (figura 25) encontramos:

Campo 1 Apartado donde se puede informar a la hora de proceder con la prueba, etc.

Campo 2 Formulario donde se selecciona la prueba a realizar y se introduce la palabra de paso.

Campo 3 Botón para proceder a validar los datos del campo 2 y acceder a la prueba. Si estos datos no son correctos, se avisa con un mensaje.

7.7.2.10 Vista del modo evaluación

Al igual que en el modo entrenamiento, se dispone de una vista (figura 26) donde poder ir respondiendo a cada ejercicio que servirá como evaluación.

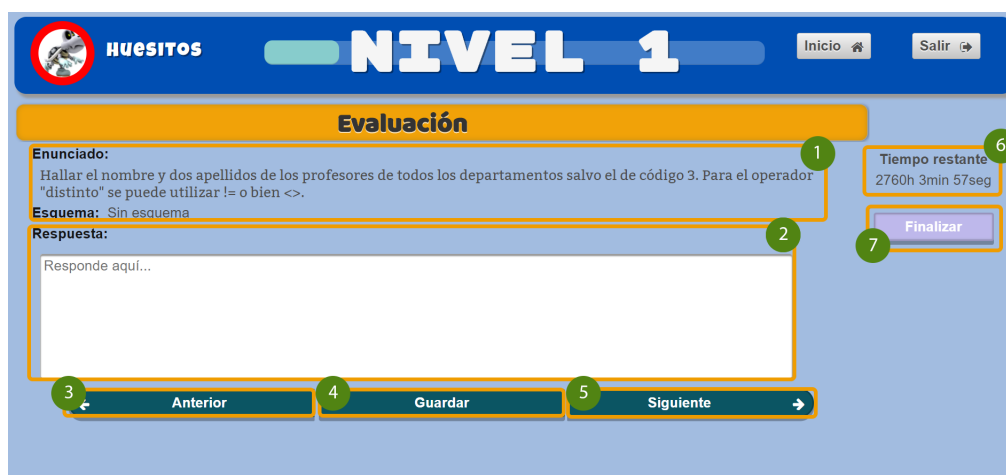


Figura 26: Vista del modo evaluación.

Esta formada por:

Campo 1 Información sobre la pregunta a contestar.

Campo 2 Campo donde el alumno introducirá su respuesta.

Campo 3 Botón para volver al ejercicio anterior.

Campo 4 Botón para guardar la respuesta en la base de datos.

Campo 5 Botón para pasar a la siguiente pregunta.

Campo 6 Tiempo restante para termina la prueba. En caso de que acabe el tiempo, se cierra automáticamente.

Campo 7 Botón donde se dará por finalizada la prueba.

7.7.2.11 Vista de la información del alumno

En esta vista (figura 27) podemos encontrar toda la información del perfil del alumno. Además, también podemos modificar algunos campos.

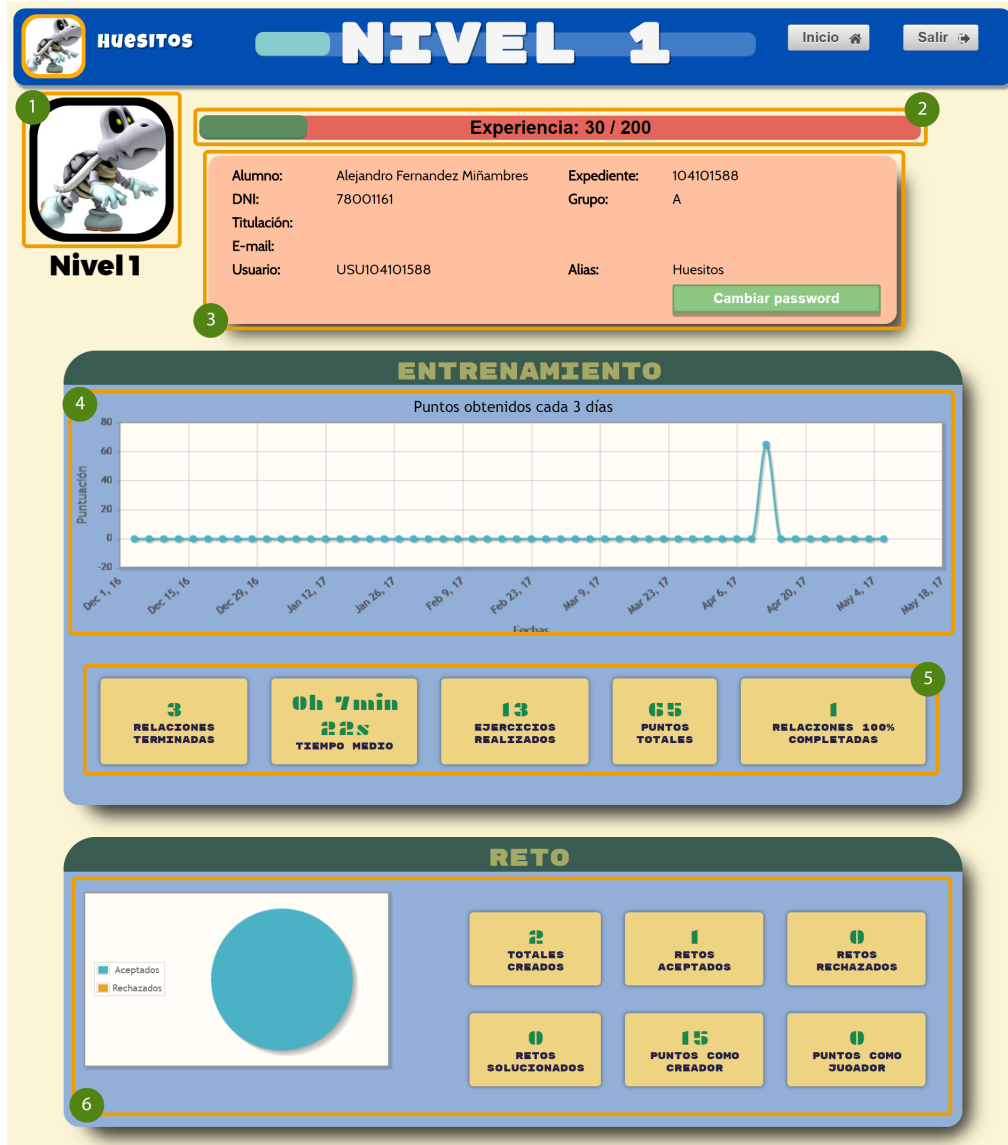


Figura 27: Vista de la información del alumno.

Se compone de:

Campo 1 Avatar del usuario. Cada usuario puede modificar su avatar si pulsa sobre la imagen. En caso de no tener, se deja el por defecto.

Campo 2 Barra de experiencia que muestra la experiencia actual y la necesaria para subir de nivel.

Campo 3 Información de alumno. El alias se puede cambiar pulsando sobre el y también permite el cambio de contraseña.

Campo 4 Gráfica que muestra la puntuación obtenida a lo largo del tiempo.

Campo 5 Diferente información referida al modo entrenamiento.

Campo 6 Diferente información referida al modo reto.

7.7.3. Interfaz del rol “profesor”

7.7.3.1 Cabera para las vistas del rol “profesor”

Al igual que en la interfaz de alumno, en esta vista (figura 28) disponemos de una cabecera común para todas las vistas.

En ella encontramos:

Campo 1 Botón inicio, para poder volver a la vista principal.

Campo 2 Botón salir, para cerrar la sesión.



Figura 28: Cabecera para las vistas del rol “profesor”.

7.7.3.2 Vista de inicio

Vista (figura 29) principal desde la cual podemos ir navegando a las diferentes vistas.

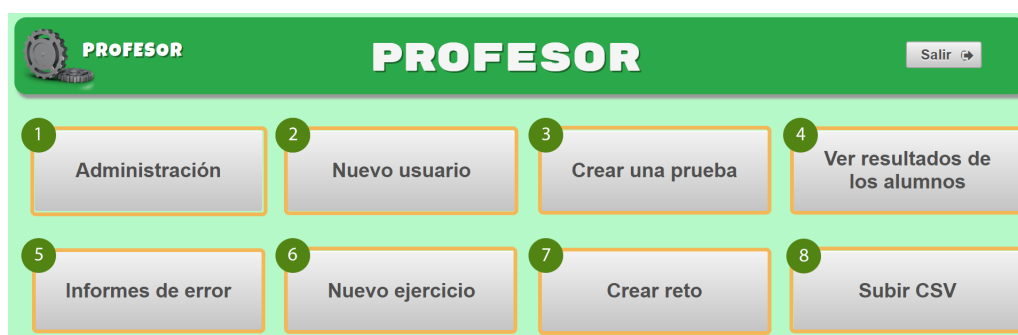


Figura 29: Vista de inicio del rol “profesor”.

Encontramos:

Campo 1 Botón para acceder a la vista de administración de alumno y ejercicios.

Campo 2 Botón para acceder a la vista donde podremos crear un nuevo usuario.

Campo 3 Botón para acceder a la vista donde podremos crear pruebas.

Campo 4 Botón para ver los resultados de los alumnos, en las diferentes pruebas, así como proceder a su corrección y cambiar la visibilidad.

Campo 5 Botón para ver lo informes de error mandados por los alumnos.

Campo 6 Botón para acceder a la vista donde podremos crear un nuevo ejercicio.

Campo 7 Botón para crear y validar retos.

Campo 8 Botón para subir acceder a la vista donde se pueden crear alumnos y ejercicios mediante archivos csv.

7.7.3.3 Vista de administración de usuarios

Esta vista (figura 30) nos permite administrar todo lo relacionado con los usuarios.

	Expediente	DNI/NIF	Nombre	Primer apellido	Segundo apellido	Grupo	Curso	Nota	
•	104103426	99999999X	MARIA	FERNANDEZ	FERNANDEZ	C	2017	1	✓
•	104101333	99999998X	DUMMY1	APDUMMY1	APDUMMY2	A	2017	2	✓
•	104101440	99999997X	DUMMY2	APDUMMY1		A	2017	6	✓
•	104101588	78001161Y	ALEJANDRO	FERNANDEZ	MIÑAMBRES	A	2017	7	✓

Figura 30: Vista de administración de usuarios.

En ella podemos ver:

Campo 1 Botón para acceder al apartado de administración de preguntas.

Campo 2 Listado de todos los alumnos, así como su información básica.

Campo 3 Campo para buscar en la lista. Filtra por varias columnas.

Campo 4 Expansión de la información, donde podemos ver información más avanzada.

Campo 5 Botón para borrar el avatar y dejarlo por defecto.

Campo 6 Botón para dejar la contraseña por defecto.

Campo 7 Botón para aplicar los cambios realizados.

7.7.3.4 Vista de administración de ejercicios

Al igual que la vista para administrar los usuarios, también disponemos el apartado para administrar los ejercicios.

En esta vista (figura 31) encontramos:

Campo 1 Listado de todos los ejercicios disponibles.

Campo 2 Campos para filtrar la lista, permite combinar los filtrados.

Campo 3 Botón para acceder a toda la información disponible de ese ejercicio e incluso permite su modificación.

Enunciado	Etiquetas	Esquema BD	Tipo
Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++	T1	BD	🔍
Hallar el nombre y dos apellidos de los profesores de todos los departamentos salvo el de código 3. Para el operador "distinto" se puede utilizar != o bien <>.	T1	BD	🔍
Hallar el nombre y dos apellidos de los profesores cuyo correo está en el servidor "lcc.uma.es". En este caso hay que utilizar el operador LIKE. Recordad que nombre LIKE 'A%' será verdadero si nombre empieza por 'A'.	T1	BD	🔍
Mostrar la lista de alumnos que no disponen de correo electrónico. Recordad el uso del operador IS NULL. Probad poniendo la expresión WHERE email = NULL .	T2	BD	🔍
Liste el nombre de las asignaturas de tercero, informando del total de créditos, de la proporción de teoría y de prácticas en tanto por ciento.	T1	BD	🔍
Muestre la lista de las notas de la asignatura 112 de la tabla MATRICULAR. Liste el código del alumno junto a su nota ordenado por el primero. Usad ORDER BY.	T2	BD	🔍
Haga la consulta 4 pero muestre la lista como: El alumno ... no dispone de Correo. Utilizad			...

Figura 31: Vista de administración de ejercicios.

7.7.3.5 Vista para crear un nuevo usuario

En la vista (figura 32) para crear un nuevo usuario, disponemos de un formulario donde debemos introducir toda la información de dicho usuario.

Encontramos:

Campo 1 Formulario para introducir la información relevante al usuario.

Campo 2 Formulario para introducir la información relevante a los datos de matriculación.

Campo 3 Botón para guardar y crear el usuario. En caso de error se avisa con un mensaje.

The screenshot shows a web interface for creating a new user profile. The header is green with the text 'PROFESOR' and 'Nuevo usuario'. There are 'Inicio' and 'Salir' buttons in the top right. The main form is divided into two sections: 'Datos del usuario' and 'Datos de matriculación'. The 'Datos del usuario' section contains several input fields: 'Expediente:', 'DNI/NIF:', 'Nombre:', 'Primer apellido:', 'Segundo apellido:', 'Teléfono:', 'E-mail:', 'Sexo*' (with a dropdown menu), 'Dirección:', 'Fecha nacimiento:', and 'Titulación:'. The 'Datos de matriculación' section contains three input fields: 'Grupo:', 'Curso:', and 'Asignatura*' (with a dropdown menu). At the bottom of the form, there are two buttons: 'Guardar' and 'Cancelar'. Three numbered callouts (1, 2, 3) are present: callout 1 points to the 'Datos de matriculación' section, callout 2 points to the 'Guardar' button, and callout 3 points to the 'Guardar' button.

Figura 32: Vista nuevo usuario.

7.7.3.6 Vista para crear una prueba

Para poder crear una prueba, necesitamos introducir cierta información.

En esta vista (figura 33) encontramos:

Campo 1 Campo para introducir el título de la prueba.

Campo 2 Campo para indicar a que grupo irá dirigida la prueba.

Campo 3 Campo para indicar la fecha de comienzo y finalización de la prueba.

Campo 4 Campo para introducir la palabra de paso.

Campo 5 Lista de todos los ejercicios disponibles.

Campo 6 Campos para filtrar los ejercicios. Permite combinar el filtrado.

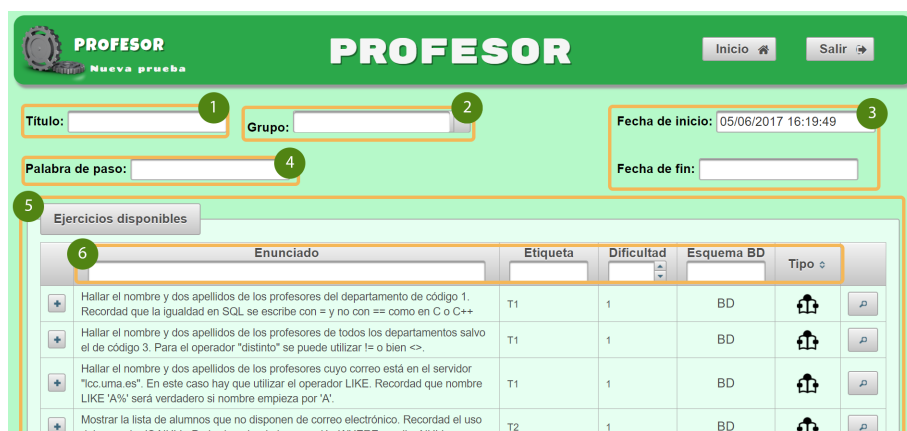


Figura 33: Vista nueva prueba.

7.7.3.7 Vista para ver las pruebas de los alumnos

Una vez que la prueba ha sido enviada para que los alumnos la resuelvan, tenemos que proceder a su corrección y verificación.

En esta vista (figura 34) encontramos:

Campo 1 Podremos ver todas las pruebas que han sido enviada y están a la espera de corrección. Por cada prueba podemos mandar al sistema que las corrija.

Campo 2 Una vez que las pruebas han sido corregida, podemos habilitar o des-habilitar su visibilidad para los alumnos.

Campo 3 Podemos ver los resultados de los alumnos en cada prueba realizada. También permite la modificación de la puntuación.

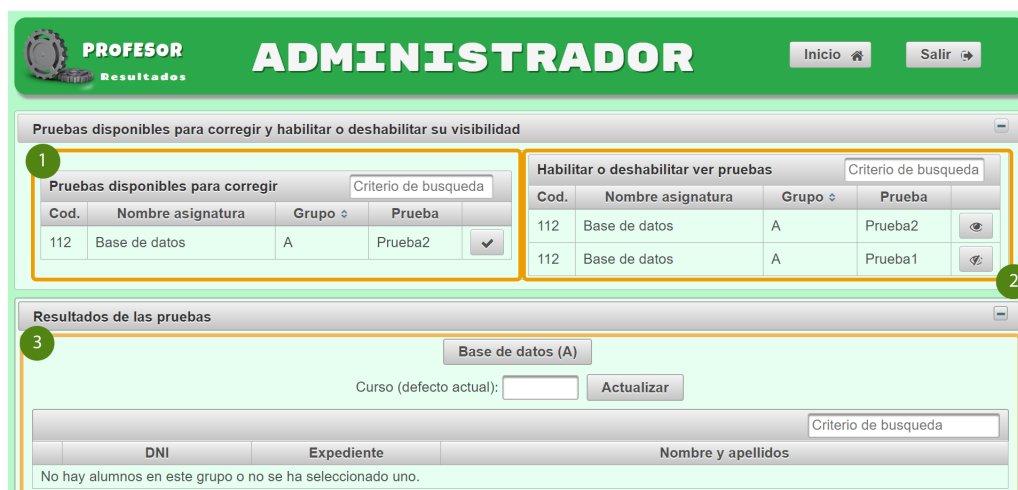


Figura 34: Vista para ver las pruebas de los alumnos.

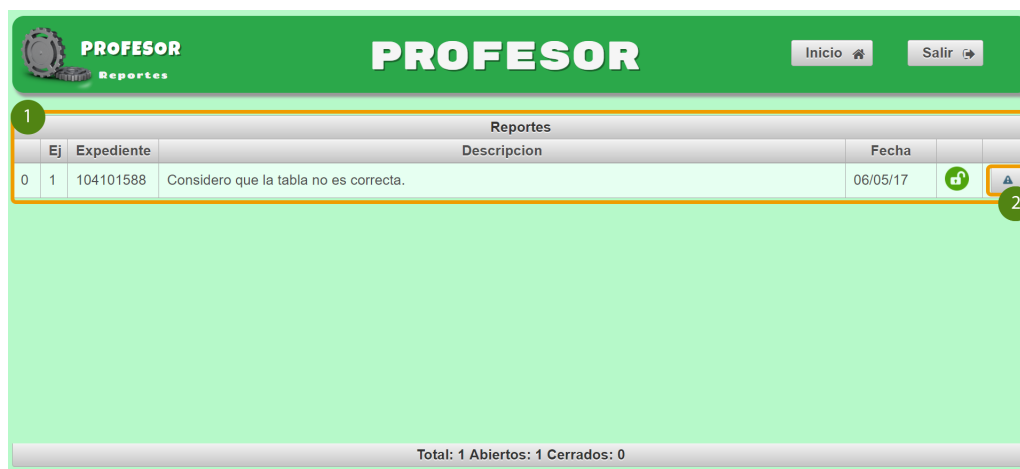
7.7.3.8 Vista para administrar los informes de los ejercicios

En esta vista (figura 35) podemos ver los diferentes informes que los alumnos han enviado. Esto informes van referidos a errores relacionados con los ejercicios.

Encontramos:

Campo 1 Listado de todos los informes enviados por los alumnos.

Campo 2 Botón que nos permite acceder directamente al ejercicio, para su modificación.



Reportes				
Ej	Expediente	Descripcion	Fecha	
0	1	104101588	Considero que la tabla no es correcta.	06/05/17

Total: 1 Abiertos: 1 Cerrados: 0

Figura 35: Vista para administrar los informes de los ejercicios.

7.7.3.9 Vista para crear un nuevo ejercicio

Al igual que tenemos una vista para crear usuarios, también tenemos una vista (figura 36) para crear los ejercicios.

En ella encontramos:

Campo 1 Campo para indicar las etiquetas de los ejercicios (separar por comas).

Campo 2 Campo para indicar la base de datos donde se corregirá el ejercicio.

Campo 3 Campo para introducir el enunciado del ejercicio.

Campo 4 Campo para indicar el esquema de la sentencia.

Campo 5 Campo para indicar la dificultad del ejercicio.

Campo 6 Campo para seleccionar el tipo de ejercicio.

Campo 7 Sección donde se añaden las soluciones.

Campo 8 Botones para guardar el ejercicio o cancelar la operación.

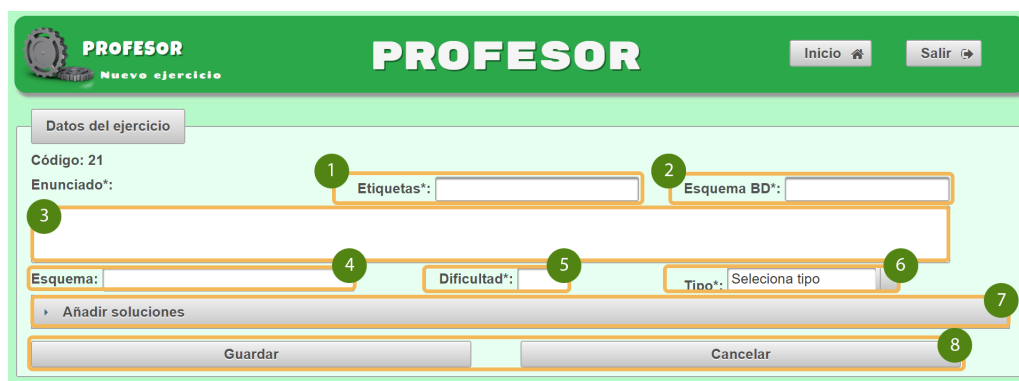


Figura 36: Vista nuevo ejercicio.

7.7.3.10 Vista donde validar los retos

Las sentencias de retos creadas por los usuarios tienen que ser validadas. En esta vista (figura 37) podemos realizar dicha tarea.

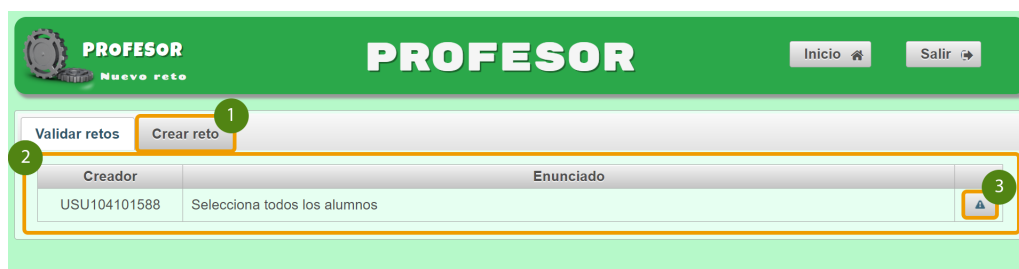


Figura 37: Vista para validar los retos.

Encontramos:

Campo 1 Botón para acceder a la vista donde crear nuevos retos.

Campo 2 Listado de todas la sentencias para retos enviadas por los alumnos, que se encuentran en estado “pendientes de validación”.

Campo 3 Botón para acceder a información detallada de la sentencia. Permite su modificación.

7.7.3.11 Vista para crear un reto

Una vez que la sentencias han sido validadas, podemos crear un reto. Los retos son conjuntos de sentencias que han enviado los alumnos.

En esta vista (figura 38) encontramos:

Campo 1 Lista de las sentencia de reto disponibles.

Campo 2 Botón que permite ver información detallada de la sentencia.

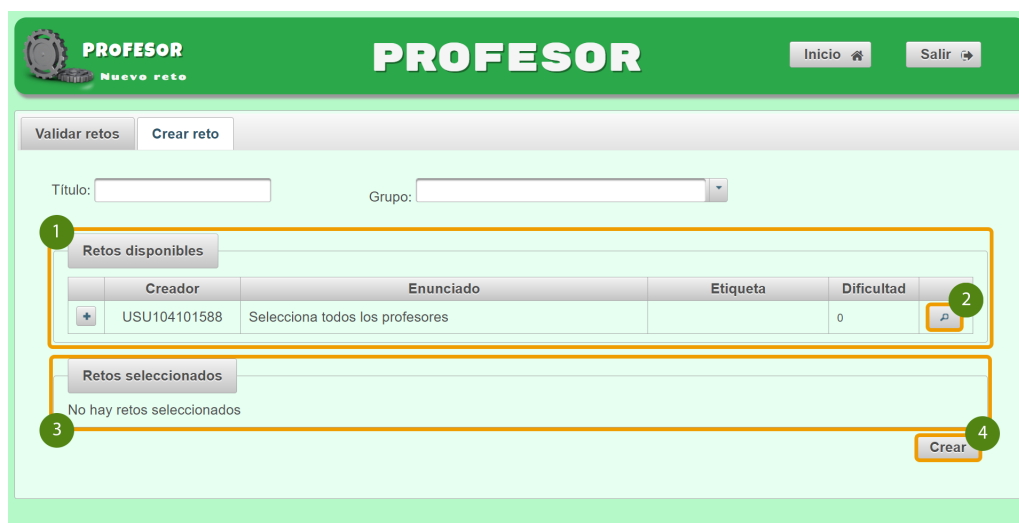


Figura 38: Vista para crear un reto.

Campo 3 Lista de las sentencias seleccionadas como candidatas.

Campo 4 Botón para crear el reto.

7.7.4. Interfaz del rol “administrador”

7.7.4.1 Cabecera para las vistas del rol “administrador”

Al igual que en todas las vistas, disponemos de una cabecera (figura 39) que es común a todas las vistas del rol administrador.

En ella encontramos:

Campo 1 Botón para cerrar la sesión.



Figura 39: Cabecera del rol administrador.

7.7.4.2 Vista de inicio del rol “administrador”

Vista (figura 40) desde la cual podrá navegar a las diferentes funciones proporcionadas.

Encontramos:

Campo 1 Botón para acceder a la parte donde se gestionan los profesores.

Campo 2 Botón para acceder a la parte donde se gestionan las asignaturas.

Campo 3 Botón para acceder a la parte donde se gestionan los procesos.

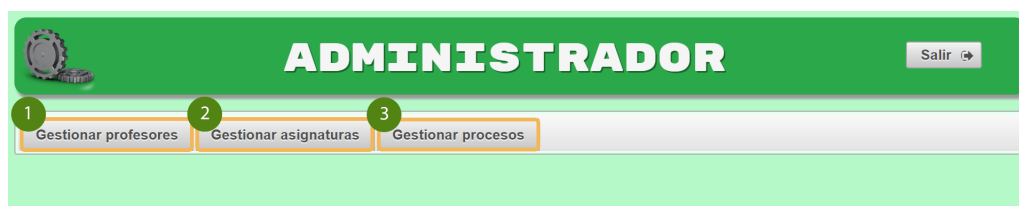


Figura 40: Vista al iniciar como administrador.

7.7.4.3 Vista de gestión de profesores

Esta vista (figura 41) nos permite gestionar todo lo relacionado con los profesores, desde dar de alta uno nuevo, asignarle que asignaturas impartirá, editar su información o darle de baja.

Podemos encontrar:



Figura 41: Vista para gestionar los profesores.

Campo 1 Listado de todos los profesores que están dados de alta en el sistema.

Campo 2 Botón para añadir un nuevo profesor.

Campo 3 Botón para borrar el profesor y darlo de baja en el sistema.

Campo 4 Botón para editar la información del profesor, así como indicar que asignaturas impartirá.

7.7.4.4 Vista de gestión de asignaturas

En esta vista (figura 42) podemos crear asignaturas que serán impartidas por los profesores.

Encontramos:

Campo 1 Listado de todas las asignaturas creadas.

Campo 2 Botón para añadir una nueva asignatura.

Campo 3 Botón para borrar una asignatura.

Campo 4 Botón para editar la información de la asignatura.



Figura 42: Vista para gestionar las asignaturas.

7.7.4.5 Vista de gestión de procesos

En esta vista (figura 43) podremos administrar los procesos.

Encontramos:

Campo 1 Listado de todos los procesos creados.

Campo 2 Botón para añadir un nuevo proceso.

Campo 3 Botón para borrar un proceso.

Campo 4 Botón para ejecutar el proceso.



Figura 43: Vista vista para gestionar los procesos.

8. Conclusión

Tras la realización de este trabajo de fin de grado, he llevado a cabo una de las tareas que considero más importantes de la vida, la cual consiste en no dejar de aprender. Muchas personas ya me habían mencionado esto en la carrera, y durante la elaboración me he dado cuenta que aparte de poder aplicar los conocimientos que ya tenía, he tenido que aprender otros para llevar a cabo los objetivos del proyecto.

Todos estos nuevos conocimientos han sido adquiridos gracias a mis tutores, búsqueda de información, apoyo de grandes comunidades formadas por desarrolladores informáticos, compañeros que en el futuro también serán ingenieros y no menos importante, aprender a base de prueba y error.

La realización de este trabajo se ha llevado a cabo por una persona, por lo tanto a la hora de escoger un camino u otro, se ha decidido bajo mi criterio, dando la posibilidad de que no sea el camino más correcto. Por el contrario, si fuera un trabajo colectivo, habría un abanico más amplio de opiniones, incluso dando las posibilidades de tomar decisiones mixtas.

Uno de los puntos en el cual invertí bastante tiempo, fue qué decisiones tomar para la elaboración de la interfaz, ya que la idea principal era hacer una interfaz fácil, útil y visualmente agradable. Nunca anteriormente me había encontrado en esta situación y tras horas de cambios y cambios, concluí con una interfaz que considero que es bastante amigable.

Y el otro punto que destacaría, es que para la realización de este proyecto no he tenido que invertir ningún tipo de recurso económico ya que la mayoría de las herramientas usadas son libres o las proporciona la Universidad de Málaga.

Como trabajos futuros, tengo que decir que la aplicación es una aplicación que acepta cualquier tipo de idea dentro de la idea global, es decir, se podría desarrollar una aplicación más completa. Por ejemplo, visualmente se podría aumentar los efectos, se podría aumentar los módulos de estadísticas e inclusive se podría expandir el sistema para corregir otro tipo de lenguajes, internacionalizar la aplicación, etc.

Referencias

- [1] Wikipedia. Ludificación, 2017. <https://es.wikipedia.org/w/index.php?title=Ludificaci%C3%B3n&oldid=98859528>. (5 Abril 2017).
- [2] Wikipedia. Derecho de acceso a internet, 2017. https://es.wikipedia.org/w/index.php?title=Derecho_de_acceso_a_Internet&oldid=94010685. (5 Abril 2017).
- [3] Johannes Schildgen. Sql island. <http://wwlgis.informatik.uni-kl.de/extra/game/>. (7 Abril 2017).
- [4] Inc Udemy. Udemy, 2017. <https://www.udemy.com/>. (15 Junio 2017).
- [5] SoloLearn Inc. SoloLearn, 2017. <https://www.sololearn.com/>. (3 Junio 2017).
- [6] Codecademy. Codecademy, 2017. <https://www.codecademy.com/>. (15 Junio 2017).
- [7] Codewars. Codewars, 2017. <https://www.codewars.com/>. (15 Junio 2017).
- [8] Wikipedia. Desarrollo ágil de software. https://es.wikipedia.org/w/index.php?title=Desarrollo_%C3%A1gil_de_software&oldid=98820458. (7 Abril 2017).
- [9] Fog Creek Software. Trello. <https://trello.com>. (9 Abril 2017).
- [10] Wikipedia. Java (lenguaje de programación), 2017. [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=99266743](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=99266743). (13 Abril 2017).
- [11] Wikipedia. Sql, 2017. <https://es.wikipedia.org/w/index.php?title=SQL&oldid=98846979>. (13 Abril 2017).
- [12] Wikipedia. Html, 2017. <https://es.wikipedia.org/w/index.php?title=HTML&oldid=98852829>. (14 Abril 2017).
- [13] W3C. W3c, 2017. <http://www.w3c.es/>. (8 Junio 2017).
- [14] Wikipedia. Hoja de estilos en cascada, 2017. https://es.wikipedia.org/w/index.php?title=Hoja_de_estilos_en_cascada&oldid=98892181. (14 Abril 2017).
- [15] Wikipedia. Javascript, 2017. <https://es.wikipedia.org/w/index.php?title=JavaScript&oldid=99182148>. (16 Abril 2017).
- [16] Wikipedia. JQuery, 2017. <https://es.wikipedia.org/w/index.php?title=jQuery&oldid=98734549>. (16 Abril 2017).
- [17] Wikipedia. Javaserver pages, 2017. https://es.wikipedia.org/w/index.php?title=JavaServer_Pages&oldid=98308952. (18 Abril 2017).

- [18] Wikipedia. Tomcat, 2017. <https://es.wikipedia.org/w/index.php?title=Tomcat&oldid=99113758>. (18 Abril 2017).
- [19] Oracle Corporation. Oracle, 2017. <https://www.oracle.com>. (18 Abril 2017).
- [20] Oracle Corporation. Netbeans ide, 2017. <https://netbeans.org/>. (20 Abril 2017).
- [21] Oracle Corporation. Sql developer data modeler, 2017. <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>. (20 Abril 2017).
- [22] Oracle Corporation. Sql developer, 2017. <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index-097090.html>. (20 Abril 2017).
- [23] Primetek. Primeface, 2017. <https://www.primefaces.org/>. (20 Abril 2017).
- [24] Dave Gandy. Font awesome, 2017. <http://fontawesome.io/>. (21 Abril 2017).
- [25] Google. Google font, 2017. <https://fonts.google.com/>. (21 Abril 2017).
- [26] Wikipedia. Lenguaje unificado de modelado, 2017. https://es.wikipedia.org/w/index.php?title=Lenguaje_unificado_de_modelado&oldid=97764494. (27 Abril 2017).
- [27] Wikipedia. Lenguaje unificado de modelado, 2017. https://es.wikipedia.org/w/index.php?title=Java_Database_Connectivity&oldid=81947156. (29 Abril 2017).

Índice de figuras

1	SQL Island.	4
2	Metodología de desarrollo ágil.	7
3	Tablón GameSQL en Trello.	8
4	Casos de uso general.	20
5	Casos de uso del actor alumno.	21
6	Casos de uso del actor profesor.	21
7	Casos de uso del actor administrador.	21
8	Modelo en tres capas.	22
9	Modelo de datos conceptual.	23
10	Modelo relacional.	24
11	Directorios.	25
12	Diagrama de Venn.	26
13	Flujo de la interfaz para el rol “alumno”.	29
14	Flujo de la interfaz para el rol “profesor”.	29
15	Flujo de la interfaz para el rol “administrador”.	30
16	Vista de inicio de sesión.	30
17	Cabecera para las vistas del rol “alumno”.	31
18	Vista de inicio del rol “alumno”.	31
19	Vista del modo entrenamiento.	32
20	Vistas al subir de nivel y aumentar puntuación.	33
21	Vista para informar de error en un ejercicio.	33
22	Vista de los retos disponibles.	34
23	Vista de las consultas creadas.	34
24	Vista para crear un consulta de reto.	35
25	Vista al iniciar una prueba.	35
26	Vista del modo evaluación.	36
27	Vista de la información del alumno.	37
28	Cabecera para las vistas del rol “profesor”.	38
29	Vista de inicio del rol “profesor”.	38
30	Vista de administración de usuarios.	39
31	Vista de administración de ejercicios.	40
32	Vista nuevo usuario.	41
33	Vista nueva prueba.	42
34	Vista para ver las pruebas de los alumnos.	42
35	Vista para administrar los informes de los ejercicios.	43
36	Vista nuevo ejercicio.	44
37	Vista para validar los retos.	44
38	Vista para crear un reto.	45
39	Cabecera del rol administrador.	45
40	Vista al iniciar como administrador.	46
41	Vista para gestionar los profesores.	46
42	Vista para gestionar las asignaturas.	47
43	Vista vista para gestionar los procesos.	47
44	Modo entrenamiento.	79

45	Tabla de estado y puntuación.	80
46	Ventana para informar de un error.	80
47	Estados disponibles (espera, valido, no valido).	81
48	Iniciar una prueba.	82
49	Ventana para resolver una prueba.	82
50	Información del alumno.	83
51	Estadísticas del modo entrenamiento.	83
52	Estadísticas del modo reto.	83
53	Cambiar alias.	84
54	Cambiar avatar.	84
55	Cambiar contraseña.	84
56	Resultados de las pruebas.	85
57	Vista para crear un nuevo alumno.	87
58	Vista para crear un nuevo ejercicio.	88
59	Vista para editar un usuario.	89
60	Vista para editar una pregunta.	90
61	Lista de informes.	90
62	Vista para crear una prueba.	91
63	Puntos creador.	91
64	Puntos del reto.	91
65	Vista para crear un reto.	92
66	Vista para ver los resultados.	92
67	Pruebas disponibles para corregir.	93
68	Habilitar o deshabilitar la visibilidad.	93
69	Resultados de las pruebas.	93
70	Subir mediante archivo CSV.	94
71	Crear profesor.	96
72	Editar y asignar una asignatura.	97
73	Crear asignatura.	98
74	Editar asignatura.	98
75	Crear proceso.	99
76	Editar proceso.	100

Anexos

Anexo I

Pruebas unitarias

A. Pruebas unitarias

En esta sección iremos comprobando una a una todas las funcionalidades que podemos encontrar en la aplicación, para determinar que funcionan correctamente.

Para su realización contaremos con una tabla (tabla 1) donde se indicará la acción a probar, los argumentos a introducir (si es necesario), la salida obtenida y su estado.

Acción	
Argumentos	
Salida	
Estado	

Tabla 1: Tabla ejemplo para las pruebas unitarias.

El flujo que seguiremos, será el mismo que se tendrá que usar para hacer funcionar la aplicación, es decir, partiendo de la base de que tenemos un usuario con rol administrador dado de alta en el sistema, iremos probando sus funcionalidades, las del rol profesor y finalmente las del rol alumno.

A.1. Pruebas unitarias: Generales

A.1.1. Inicio de sesión con la base de datos desconectada

Acción	Iniciar la sesión con la base de datos apagada o sin conexión.
Argumentos	Nombre de usuario y contraseña.
Salida	No hay conexión con la BD.
Estado	Correcto ✓

Prueba unitaria 1: General - Iniciar sesión sin conexión con la BD.

A.1.2. Iniciar sesión con parámetros incorrectos

Acción	Iniciar sesión con parámetros incorrectos o inexistentes.
Argumentos	Nombre de usuario y contraseña incorrectos.
Salida	El usuario o contraseña es incorrecto.
Estado	Correcto ✓

Prueba unitaria 2: General - Iniciar sesión con parámetros incorrectos.

A.2. Pruebas unitarias: rol administrador

A.2.1. Inicio de sesión

Acción	Iniciar la sesión como administrador.
Argumentos	Nombre de usuario y contraseña.
Salida	Sesión iniciada.
Estado	Correcto ✓

Prueba unitaria 3: Rol administrador - Inicio sesión.

A.2.2. Añadir un profesor

Acción	Crear un nuevo profesor y dar alta en el sistema.
Argumentos	NRP, Id usuario, Nombre, Primer apellido, Segundo apellido, Sexo, E-mail, Departamento, Fecha nacimiento, Antigüedad y Contraseña.
Salida	Añadido a la base de datos y dado de alta.
Estado	Correcto ✓

Prueba unitaria 4: Rol administrador - Crear profesor.

A.2.3. Añadir un profesor que ya existe

Acción	Añadir un profesor cuyos datos ya existen en la base de datos.
Argumentos	NRP, Id usuario, Nombre, Primer apellido, Segundo apellido, Sexo, E-mail, Departamento, Fecha nacimiento, Antigüedad y Contraseña.
Salida	NRP o Id usuario ya existe.
Estado	Correcto ✓

Prueba unitaria 5: Rol administrador - Añadir un profesor ya existente.

A.2.4. Añadir un profesor con los campos obligatorios vacíos

Acción	Añadir un profesor con los campos obligatorios vacíos.
Argumentos	Sin argumentos obligatorios.
Salida	Campos obligatorios vacíos.
Estado	Correcto ✓

Prueba unitaria 6: Rol administrador - Añadir un profesor con los campos obligatorios vacíos.

A.2.5. Añadir impartición de asignatura a un profesor

Acción	Añadir asignatura que imparte.
Argumentos	Turno, Código de la asignatura y Curso.
Salida	Añadida a la base de datos.
Estado	Correcto ✓

Prueba unitaria 7: Rol administrador - Añadir asignatura que impartirá un profesor.

A.2.6. Añadir impartición de asignatura que no existe a un profesor

Acción	Añadir una asignatura que no existe a un profesor.
Argumentos	Turno, Código de la asignatura (no existe) y Curso.
Salida	Error al añadir.
Estado	Correcto ✓

Prueba unitaria 8: Rol administrador - Añadir asignatura no existente a un profesor.

A.2.7. Añadir impartición de asignatura con los campos obligatorios vacíos

Acción	Añadir una asignatura a un profesor con los campos obligatorios vacíos.
Argumentos	Sin argumentos obligatorios.
Salida	Campos obligatorios vacíos.
Estado	Correcto ✓

Prueba unitaria 9: Rol administrador - Añadir impartición con los campos obligatorios vacíos.

A.2.8. Borrar impartición de asignatura

Acción	Eliminar la impartición de una asignatura en un profesor.
Argumentos	Sin argumentos.
Salida	Borrado correctamente.
Estado	Correcto ✓

Prueba unitaria 10: Rol administrador - Borrar impartición de asignatura.

A.2.9. Editar la información de un profesor

Acción	Editar la información de un profesor que ya existe.
Argumentos	Información a editar.
Salida	Editado correctamente.
Estado	Correcto ✓

Prueba unitaria 11: Rol administrador - Editar la información de un profesor.

A.2.10. Borrar un profesor

Acción	Borrar un profesor y dar de baja.
Argumentos	Sin argumentos.
Salida	Borrado correctamente.
Estado	Correcto ✓

Prueba unitaria 12: Rol administrador - Borrar un profesor.

A.2.11. Añadir asignatura

Acción	Añadir una nueva asignatura.
Argumentos	Código, nombre, cuatrimestre, valor mínimo para aprobar, esquema bd.
Salida	Añadido correctamente.
Estado	Correcto ✓

Prueba unitaria 13: Rol administrador - Añadir una asignatura.

A.2.12. Añadir asignatura que ya existe

Acción	Añadir una nueva asignatura, con el mismo código.
Argumentos	Código (ya existe), nombre, cuatrimestre, valor mínimo para aprobar, esquema bd.
Salida	Asignatura con ese código ya existe.
Estado	Correcto ✓

Prueba unitaria 14: Rol administrador - Añadir una asignatura que ya existe.

A.2.13. Añadir asignatura sin los campos obligatorios

Acción	Añadir una asignatura sin los campos obligatorios.
Argumentos	Sin argumentos obligatorios.
Salida	Campos obligatorios vacíos.
Estado	Correcto ✓

Prueba unitaria 15: Rol administrador - Añadir asignatura sin los campos obligatorios.

A.2.14. Editar asignatura

Acción	Editar la información de una asignatura.
Argumentos	Información a editar.
Salida	Editado correctamente.
Estado	Correcto ✓

Prueba unitaria 16: Rol administrador - Editar asignatura.

A.2.15. Borrar una asignatura

Acción	Borrar una asignatura.
Argumentos	Sin argumentos.
Salida	Borrado correctamente.
Estado	Correcto ✓

Prueba unitaria 17: Rol administrador - Borrar una asignatura.

A.2.16. Borrar una asignatura con impartición o matriculación

Acción	Borrar una asignatura, la cual está impartida por un profesor y tiene alumnos matriculados.
Argumentos	Sin argumentos.
Salida	Revisa que no tenga usuarios asignados.
Estado	Correcto ✓

Prueba unitaria 18: Rol administrador - Borrar una asignatura con impartición o matriculación.

A.2.17. Añadir un proceso

Acción	Añadir un proceso.
Argumentos	Proceso, argumentos, esquemaBD.
Salida	Creado correctamente.
Estado	Correcto ✓

Prueba unitaria 19: Rol administrador - Añadir un proceso.

A.2.18. Añadir un proceso que ya existe

Acción	Añadir un proceso, que ya existe.
Argumentos	Proceso (repetido), argumentos, esquemaBD.
Salida	El proceso no ha sido añadido.
Estado	Correcto ✓

Prueba unitaria 20: Rol administrador - Añadir un proceso.

A.2.19. Añadir un proceso sin los campos obligatorios

Acción	Añadir un proceso sin los campos obligatorios.
Argumentos	Sin argumentos obligatorios.
Salida	Campos obligatorios vacíos.
Estado	Correcto ✓

Prueba unitaria 21: Rol administrador - Añadir un proceso sin los campos obligatorios.

A.2.20. Ejecutar un proceso

Acción	Ejecutar un proceso.
Argumentos	Sin argumentos.
Salida	Ejecutado correctamente.
Estado	Correcto ✓

Prueba unitaria 22: Rol administrador - Ejecutar un proceso.

A.2.21. Ejecutar un proceso que no se encuentra en la base de datos

Acción	Ejecutar un proceso, cuyo procedimiento no se encuentra en la base de datos.
Argumentos	Argumentos del proceso.
Salida	No se ha ejecutado el proceso.
Estado	Correcto ✓

Prueba unitaria 23: Rol administrador - Ejecutar un proceso que no se encuentra en la base de datos.

A.2.22. Borrar un proceso

Acción	Borrar un proceso.
Argumentos	Sin argumentos.
Salida	Borrado correctamente.
Estado	Correcto ✓

Prueba unitaria 24: Rol administrador - Borrar un proceso.

A.3. Pruebas unitarias: rol profesor

A.3.1. Inicio de sesión

Acción	Iniciar la sesión como profesor.
Argumentos	Nombre de usuario y contraseña.
Salida	Sesión iniciada.
Estado	Correcto ✓

Prueba unitaria 25: Rol profesor - Inicio sesión.

A.3.2. Crear un alumno

Acción	Crear un nuevo alumno.
Argumentos	Expediente, DNI, nombre, primer y segundo apellido, teléfono, e-mail, sexo, dirección, fecha nacimiento, titulación, grupo, curso y asignatura.
Salida	Añadido correctamente.
Estado	Correcto ✓

Prueba unitaria 26: Rol profesor - Crear un alumno.

A.3.3. Crear un alumno ya existente

Acción	Crear un nuevo alumno ya existente.
Argumentos	Expediente, DNI, nombre, primer y segundo apellido, teléfono, e-mail, sexo, dirección, fecha nacimiento, titulación, grupo, curso y asignatura.
Salida	Ya existe un alumno con los mismos datos.
Estado	Correcto ✓

Prueba unitaria 27: Rol profesor - Crear un alumno ya existente.

A.3.4. Crear un alumno sin los campos obligatorios

Acción	Crear un nuevo alumno con los campos obligatorios vacíos.
Argumentos	Argumentos obligatorios vacíos.
Salida	Mensaje de error por cada campo no introducido.
Estado	Correcto ✓

Prueba unitaria 28: Rol profesor - Crear un alumno sin los campos obligatorios.

A.3.5. Editar información de un alumno

Acción	Editar la información de un alumno.
Argumentos	Información a editar.
Salida	Editado correctamente.
Estado	Correcto ✓

Prueba unitaria 29: Rol profesor - Editar un alumno.

A.3.6. Resetear la contraseña de un alumno

Acción	Resetear la contraseña.
Argumentos	Sin argumentos.
Salida	Reseteada correctamente.
Estado	Correcto ✓

Prueba unitaria 30: Rol profesor - Resetear contraseña de un alumno.

A.3.7. Borrar avatar de un alumno

Acción	Borrar el avatar de un alumno.
Argumentos	Sin argumentos.
Salida	Borrado correctamente.
Estado	Correcto ✓

Prueba unitaria 31: Rol profesor - Borrar avatar de un alumno.

A.3.8. Crear un ejercicio

Acción	Crear un ejercicio.
Argumentos	Etiquetas, esquema BD, enunciado, esquema, dificultad, tipo, solución y puntuación.
Salida	Añadido correctamente.
Estado	Correcto ✓

Prueba unitaria 32: Rol profesor - Crear un ejercicio.

A.3.9. Crear un ejercicio sin los campos obligatorios

Acción	Crear un ejercicio con los campos obligatorios vacíos.
Argumentos	Argumentos obligatorios vacíos.
Salida	Mensaje de error por cada argumento vacío.
Estado	Correcto ✓

Prueba unitaria 33: Rol profesor - Crear un ejercicio sin los campos obligatorios.

A.3.10. Editar un ejercicio

Acción	Editar un ejercicio.
Argumentos	Información a editar.
Salida	Editado correctamente.
Estado	Correcto ✓

Prueba unitaria 34: Rol profesor - Editar un ejercicio.

A.3.11. Cerrar un informe de error

Acción	Cerrar un informe de error.
Argumentos	Sin argumentos.
Salida	Cerrado correctamente.
Estado	Correcto ✓

Prueba unitaria 35: Rol profesor - Cerrar un informe.

A.3.12. Crear una prueba

Acción	Crear una prueba.
Argumentos	Título, grupo, fecha inicio, fecha fin, palabra de paso y ejercicios seleccionados.
Salida	Creado correctamente.
Estado	Correcto ✓

Prueba unitaria 36: Rol profesor - Crear una prueba.

A.3.13. Crear una prueba con el mismo título

Acción	Crear una prueba con un título ya existente.
Argumentos	Título (ya existe), grupo, fecha inicio, fecha fin, palabra de paso y ejercicios seleccionados.
Salida	Título de la relación ya existe.
Estado	Correcto ✓

Prueba unitaria 37: Rol profesor - Crear una prueba con el mismo título.

A.3.14. Crear una prueba sin los argumentos obligatorios

Acción	Crear una prueba sin los argumentos obligatorios.
Argumentos	Argumentos obligatorios vacíos.
Salida	Mensaje de error por cada argumento vacío.
Estado	Correcto ✓

Prueba unitaria 38: Rol profesor - Crear una prueba sin los argumentos obligatorios.

A.3.15. Corregir una prueba

Acción	Corregir una prueba.
Argumentos	Prueba a corregir.
Salida	Corregida correctamente.
Estado	Correcto ✓

Prueba unitaria 39: Rol profesor - Corregir una prueba.

A.3.16. Editar puntuación de un alumno

Acción	Editar la puntuación de una sentencia de un alumno.
Argumentos	Nueva puntuación.
Salida	Editado correctamente.
Estado	Correcto ✓

Prueba unitaria 40: Rol profesor - Editar puntuación de un alumno.

A.3.17. Habilitar/deshabilitar visibilidad de una prueba

Acción	Activar o desactivar la visibilidad de una prueba.
Argumentos	Prueba para cambiar la visibilidad.
Salida	Visibilidad cambiada correctamente.
Estado	Correcto ✓

Prueba unitaria 41: Rol profesor - Cambiar visibilidad de una prueba.

A.3.18. Filtrar pruebas por curso y asignatura

Acción	Filtrar pruebas por curso y asignatura.
Argumentos	Curso (curso actual por defecto).
Salida	Listado correctamente.
Estado	Correcto ✓

Prueba unitaria 42: Rol profesor - Filtrar pruebas por curso y asignatura.

A.3.19. Validar un reto

Acción	Validar un reto creado por un alumno.
Argumentos	Puntuación para el creador y puntuación para el reto.
Salida	Validado correctamente.
Estado	Correcto ✓

Prueba unitaria 43: Rol profesor - Validar un reto.

A.3.20. Validar un reto sin puntuar al creador

Acción	Validar un reto creado por un alumno, sin introducir la puntuación para el creador.
Argumentos	Campo puntos para el creador vacío.
Salida	No se ha puntuado al creador.
Estado	Correcto ✓

Prueba unitaria 44: Rol profesor - Validar un reto sin puntuar al creador.

A.3.21. Validar un reto sin puntuar la solución

Acción	Validar un reto creado por un alumno, sin puntuar la solución.
Argumentos	Campo puntos para la solución vacío.
Salida	Por favor, puntúa la solución.
Estado	Correcto ✓

Prueba unitaria 45: Rol profesor - Validar un reto sin puntuar al creador.

A.3.22. Crear un reto

Acción	Crear un reto
Argumentos	Título, grupo y conjunto de sentencias de retos.
Salida	Creado correctamente.
Estado	Correcto ✓

Prueba unitaria 46: Rol profesor - Crear un reto.

A.3.23. Crear un reto ya existente

Acción	Crear un reto con un título ya existente.
Argumentos	Título (ya existe), grupo y conjunto de sentencias de retos.
Salida	El título ya existe.
Estado	Correcto ✓

Prueba unitaria 47: Rol profesor - Crear un reto ya existente.

A.3.24. Crear un reto sin los argumentos obligatorios

Acción	Crear un reto sin los argumentos obligatorios.
Argumentos	Sin argumentos obligatorios.
Salida	Parámetros obligatorios vacíos.
Estado	Correcto ✓

Prueba unitaria 48: Rol profesor - Crear un reto sin los argumentos obligatorios.

A.3.25. Subir ejercicios o alumnos mediante un archivo csv

Acción	Subir ejercicios o alumnos mediante un archivo csv.
Argumentos	Archivo csv.
Salida	Script del estado.
Estado	Correcto ✓

Prueba unitaria 49: Rol profesor - Subir alumnos o ejercicios mediante csv.

A.4. Pruebas unitarias: rol alumno

A.4.1. Inicio de sesión

Acción	Iniciar la sesión como alumno.
Argumentos	Nombre de usuario y contraseña.
Salida	Sesión iniciada.
Estado	Correcto ✓

Prueba unitaria 50: Rol profesor - Inicio sesión.

A.4.2. Iniciar el modo entrenamiento

Acción	Iniciar el modo entrenamiento.
Argumentos	Sin argumentos.
Salida	Iniciado correctamente y relación generada.
Estado	Correcto ✓

Prueba unitaria 51: Rol alumno - Iniciar el modo entrenamiento.

A.4.3. Navegación por el modo entrenamiento

Acción	Navegar por los diferentes ejercicios que forman el modo entrenamiento.
Argumentos	Sin argumentos.
Salida	Navegación correcta.
Estado	Correcto ✓

Prueba unitaria 52: Rol alumno - Navegación por el modo entrenamiento.

A.4.4. Informar de un error en un ejercicio

Acción	Informar de un error en un ejercicio.
Argumentos	Descripción del error.
Salida	Informado con éxito.
Estado	Correcto ✓

Prueba unitaria 53: Rol alumno - Informar de un error en un ejercicio.

A.4.5. Responder una sentencia de entrenamiento

Acción	Responder una sentencia del modo entrenamiento.
Argumentos	Respuesta.
Salida	Respondido, validado correctamente y puntuado.
Estado	Correcto ✓

Prueba unitaria 54: Rol alumno - Responder una sentencia de entrenamiento.

A.4.6. Responder una sentencia de entrenamiento con la respuesta vacía

Acción	Responder una sentencia con la respuesta vacía.
Argumentos	Sin respuesta.
Salida	No se ha respondido.
Estado	Correcto ✓

Prueba unitaria 55: Rol alumno - Responder una sentencia de entrenamiento con la respuesta vacía.

A.4.7. Responder a una sentencia de entrenamiento con una respuesta no permitida

Acción	Responder una sentencia con palabras bloqueadas o sin estructura de sentencia SQL.
Argumentos	Respuesta incorrecta.
Salida	Palabras no permitidas o sentencia mal construida.
Estado	Correcto ✓

Prueba unitaria 56: Rol alumno - Responder a una sentencia de entrenamiento con una respuesta no permitida.

A.4.8. Validar bloque en modo entrenamiento

Acción	Validar un bloque de ejercicios en el modo entrenamiento.
Argumentos	Sin argumentos.
Salida	Suma de la puntuación obtenida.
Estado	Correcto ✓

Prueba unitaria 57: Rol alumno - Validar bloque en modo entrenamiento.

A.4.9. Aumento de experiencia

Acción	Aumento de experiencia tras validar un bloque de ejercicios en el modo entrenamiento.
Argumentos	Sin argumentos.
Salida	Experiencia aumentada.
Estado	Correcto ✓

Prueba unitaria 58: Rol alumno - Aumento de experiencia.

A.4.10. Aumento de nivel

Acción	Aumento del nivel al pasar la experiencia máxima por nivel.
Argumentos	Sin argumentos.
Salida	Nivel aumentado.
Estado	Correcto ✓

Prueba unitaria 59: Rol alumno - Aumento de nivel.

A.4.11. Crear consulta para reto

Acción	Crear una consulta para reto.
Argumentos	Enunciado, solución y esquema.
Salida	Creado con éxito.
Estado	Correcto ✓

Prueba unitaria 60: Rol alumno - Crear consulta para reto.

A.4.12. Eliminar consulta de reto creada

Acción	Eliminar consulta de reto creada.
Argumentos	Sin argumentos.
Salida	Eliminado correctamente.
Estado	Correcto ✓

Prueba unitaria 61: Rol alumno - Eliminar reto creado.

A.4.13. Responder a una consulta de reto

Acción	Responder a una consulta de reto.
Argumentos	Respuesta.
Salida	Validación y puntuación.
Estado	Correcto ✓

Prueba unitaria 62: Rol alumno - Responder a una consulta de reto.

A.4.14. Responder a una consulta de reto con la respuesta vacía

Acción	Responder a una consulta de reto con la respuesta vacía
Argumentos	Sin respuesta.
Salida	No se ha respondido.
Estado	Correcto ✓

Prueba unitaria 63: Rol alumno - Responder a una consulta de reto con la respuesta vacía.

A.4.15. Responder a una sentencia de reto con una respuesta no permitida

Acción	Responder una sentencia con palabras bloqueadas o sin estructura de sentencia SQL.
Argumentos	Respuesta incorrecta.
Salida	Palabras no permitidas o sentencia mal construida.
Estado	Correcto ✓

Prueba unitaria 64: Rol alumno - Responder a una sentencia de reto con una respuesta no permitida.

A.4.16. Acceso a una prueba

Acción	Iniciar una prueba.
Argumentos	Prueba y palabra de paso.
Salida	Acceso a la prueba.
Estado	Correcto ✓

Prueba unitaria 65: Rol alumno - Acceso a una prueba.

A.4.17. Guardar una sentencia de prueba

Acción	Guardar una sentencia de prueba.
Argumentos	Respuesta.
Salida	Guardado.
Estado	Correcto ✓

Prueba unitaria 66: Rol alumno - Guardar una sentencia de prueba.

A.4.18. Guardar una sentencia de prueba vacía

Acción	Guardar una sentencia vacía.
Argumentos	Sin respuesta.
Salida	No se puede guardar una respuesta vacía.
Estado	Correcto ✓

Prueba unitaria 67: Rol alumno - Guardar una sentencia de prueba vacía.

A.4.19. Guardar una sentencia de prueba no permitida

Acción	Guardar una sentencia con palabras bloqueadas o sin estructura de sentencia SQL.
Argumentos	Respuesta incorrecta.
Salida	Palabras no permitidas o toda sentencia debe empezar por “select”.
Estado	Correcto ✓

Prueba unitaria 68: Rol alumno - Guardar una sentencia de prueba no permitida.

A.4.20. Navegación por la prueba

Acción	Navegar por los diferentes ejercicios que forman la prueba.
Argumentos	Sin argumentos.
Salida	Navegación correcta.
Estado	Correcto ✓

Prueba unitaria 69: Rol alumno - Navegación por la prueba.

A.4.21. Finalizar una prueba

Acción	Finalizar una prueba.
Argumentos	Sin argumentos.
Salida	Salir de la prueba.
Estado	Correcto ✓

Prueba unitaria 70: Rol alumno - Finalizar un prueba.

A.4.22. Cambiar o borrar avatar

Acción	Cambiar o borrar el avatar del usuario.
Argumentos	Imagen del avatar.
Salida	Cambiado correctamente.
Estado	Correcto ✓

Prueba unitaria 71: Rol alumno - Cambiar o borrar avatar.

A.4.23. Cambiar avatar superando el espacio permitido

Acción	Cambiar el avatar, con un tamaño mayor a <tamaño definido>.
Argumentos	Imagen del avatar.
Salida	La imagen pesa más de <tamaño definido>.
Estado	Correcto ✓

Prueba unitaria 72: Rol alumno - Cambiar avatar superando el espacio permitido.

A.4.24. Cambiar contraseña

Acción	Cambiar la contraseña.
Argumentos	Introducir y repetir la nueva contraseña.
Salida	Cambiado correctamente.
Estado	Correcto ✓

Prueba unitaria 73: Rol alumno - Cambiar contraseña.

A.4.25. Cambiar contraseña con fallo de repetición

Acción	Cambiar la contraseña, introduciendo la repetición errónea.
Argumentos	Contraseña y repetición errónea.
Salida	Las claves no coinciden.
Estado	Correcto ✓

Prueba unitaria 74: Rol alumno - Cambiar contraseña con fallo de repetición.

A.4.26. Cambiar alias

Acción	Cambiar alias.
Argumentos	Nuevo alias.
Salida	Alias cambiado con éxito.
Estado	Correcto ✓

Prueba unitaria 75: Rol alumno - Cambiar alias.

A.4.27. Cambiar alias por uno ya existente

Acción	Cambiar alias, por uno que ya esta asignado a un usuario.
Argumentos	Nuevo alias.
Salida	Este alias ya existe.
Estado	Correcto ✓

Prueba unitaria 76: Rol alumno - Cambiar alias por uno ya existente.

A.4.28. Ver resultados de las pruebas

Acción	Ver los resultados de las pruebas realizadas.
Argumentos	Sin argumentos.
Salida	Sin salida.
Estado	Correcto ✓

Prueba unitaria 77: Rol alumno - Ver resultados de las pruebas.

Anexo II

Manual del alumno

B. Manual del alumno

B.1. Modo entrenamiento

En el modo entrenamiento usted podrá ir solucionando las sentencias propuestas para poner a prueba sus conocimientos.

Para acceder a este modo, pulse el botón



que le llevará a la vista (figura 44)

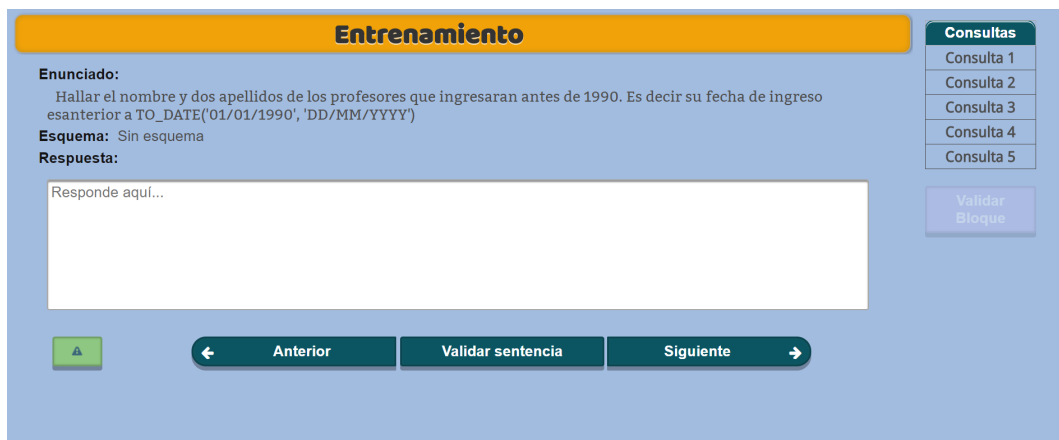


Figura 44: Modo entrenamiento.

en la cual dispondrá de:

- **Enunciado:** Enunciado que indica la sentencia que debe construir.
- **Esquema:** Esquema que indica la estructura de la sentencia.
- **Respuesta:** Campo que deberá rellenar con su solución.

Una vez rellenada la solución, debe pulsar el botón **Validar sentencia** para que la sentencia se corrija y obtener la puntuación.

Una vez corregida, el estado y la puntuación de cada sentencia se puede ver en la tabla de la derecha si mantiene el puntero encima (figura 45).

Consultas	
	Consulta 1
2.0/2.0 puntos	Consulta 2
	Consulta 3
	Consulta 4
	Consulta 5

Figura 45: Tabla de estado y puntuación.

Para navegar por las diferentes sentencias deberá pulsar los botones





Una vez obtenido un porcentaje de la puntuación máxima que puede obtener con ese conjunto de sentencias, podrá obtener la puntuación para aumentar la experiencia y cambiar de ejercicios. Para ello debe pulsar el botón



B.1.1. Informa de un error en el ejercicio

Si usted considera que el ejercicio tiene un error, puede comunicarlo al profesor mediante un formulario.

Para acceder a él debe pulsar el botón , con el cual se le abrirá una ventana (figura 46). Tendrá que rellenar el cuadro con un mensaje descriptivo del error y finalmente pulsar sobre el botón .

Reporte

Usuario: USU10

Breve descripción del problema

Cerrar

Informar

Figura 46: Ventana para informar de un error.

B.2. Modo reto

En el modo reto, usted podrá crear sentencias para retos que después de ser validadas por el profesor pueden formar parte de retos. Para acceder a este modo, debe pulsar el botón

Reto

B.2.1. Crear una nueva consulta para reto

Para crear una nueva consulta para el reto, debe pulsar los botones

Administración de consultas para retos → Nueva consulta para reto

Donde tendrá que rellenar los campos enunciado (obligatorio), solución (obligatorio) y esquema.

Una vez enviada, en el apartado

Administración de consultas para retos → Consultas para retos creadas

aparecerá la nueva consulta creada con su correspondiente estado (figura 47)




Figura 47: Estados disponibles (espera, valido, no valido).

B.2.2. Solucionar un reto

Si estamos en la sección


Retos disponibles

veremos una lista con los retos disponibles. Para solucionarlo, tendrá que pulsar sobre el botón .

B.3. Prueba evaluable

Para acceder a una prueba, deberá pulsar el botón

Prueba evaluable

que le llevará a la vista (figura 48) donde tendrá que seleccionar la prueba, introducir la palabra de paso y pulsar el botón .

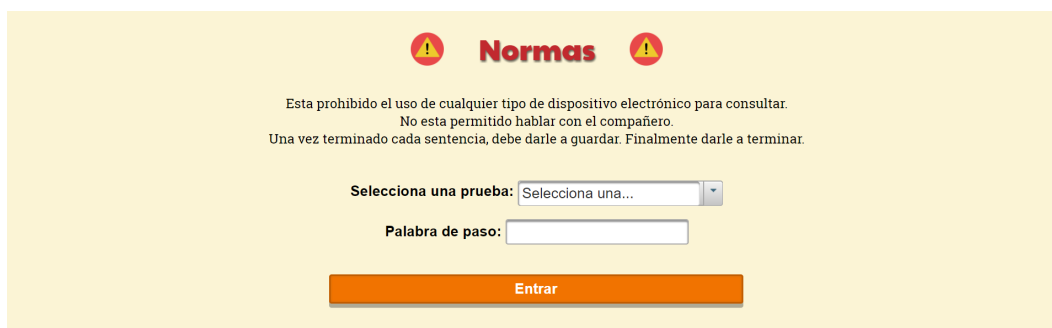


Figura 48: Iniciar una prueba.

Una vez que ha entrado ira a una nueva vista (figura 49), donde dispondrá de un tiempo.

Por cada solución que proporcione debe pulsar el botón **Guardar** y finalmente cuando considere que ha terminado la prueba, pulse el botón **Finalizar**.

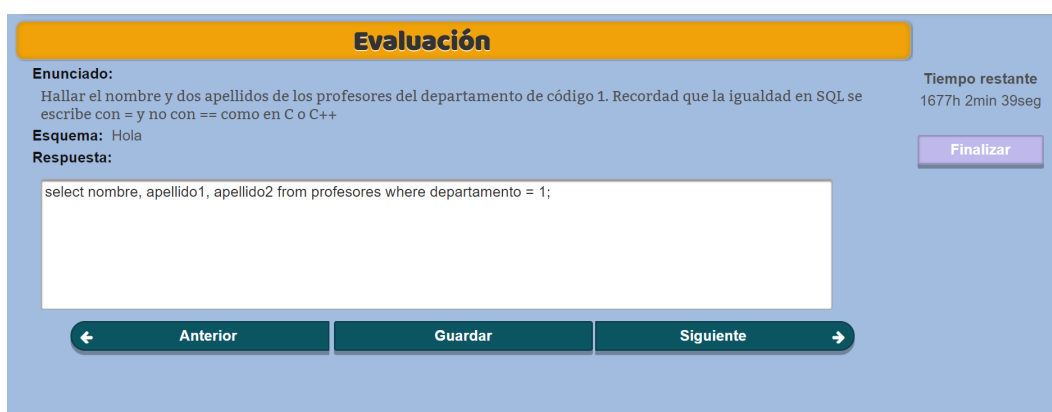


Figura 49: Ventana para resolver una prueba.

B.4. Información

Para acceder a su información, cambiar el avatar, alias o la contraseña debe pulsar sobre el botón



y accederá a la vista donde podemos encontrar diferentes apartados de como:

Información como alumno

Alumno:	Alejandro Fernandez Miñambres	Expediente:	3
DNI:	11111111N	Grupo:	A
Titulación:	Ing.Informatica		
E-mail:	ale1@ejemplo.com		
Usuario:	USU3	Alias:	Huesitos

[Cambiar password](#)

Figura 50: Información del alumno.

Estadísticas del modo entrenamiento



Figura 51: Estadísticas del modo entrenamiento.

Estadísticas del modo reto



Figura 52: Estadísticas del modo reto.

B.4.1. Cambiar alias

Si desea cambiar el alias, dentro de la información como alumno (figura 53), pulse sobre el alias actual y te dejará cambiarlo.

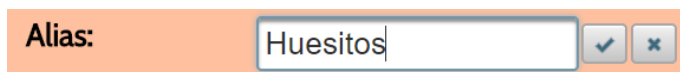


Figura 53: Cambiar alias.

B.4.2. Cambiar avatar

Si desea cambia el avatar, debe pulsar sobre el avatar actual



y se le abrirá una ventana donde cambiarlo (figura 54).

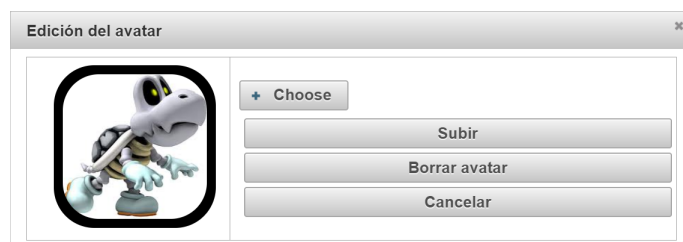


Figura 54: Cambiar avatar.

B.4.3. Cambiar contraseña

Para cambiar la contraseña, debe pulsar el botón **Cambiar password** y se le abrirá una ventana (figura 55) donde tiene que introducir y repetir la nueva. Finalmente pulsar sobre el botón **Cambiar**.

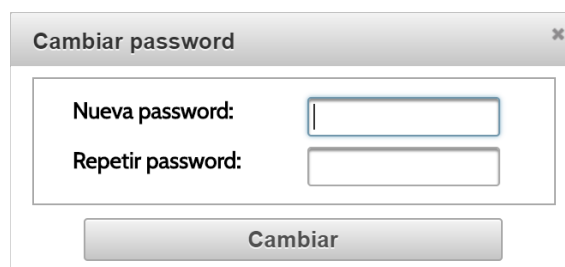


Figura 55: Cambiar contraseña.

B.5. Resultados de las pruebas

Para ver el resultado de las pruebas que ha realizado, debe pulsar el botón

Resultados de las pruebas

y accederá a una vista (figura 56) donde tendrá una lista con las pruebas realizadas y su puntuación.

Examen		Terminado a tiempo	Nota
PruebaEjemplo_USU10		Si	10.0 / 10.0
Enunciado	Tu solución	Descripcion	Puntos
Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++	<pre>select nombre, apellido1, apellido2 from profesores where departamento = 1;</pre>		5.0
Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++	<pre>select nombre, apellido1, apellido2 from profesores where departamento = 1;</pre>		5.0
Mostrar la lista de alumnos que no disponen de correo electrónico. Recordad el uso del operador IS NULL. Probad poniendo la expresión WHERE email = NULL .	<pre>select 'El alumno ' nombre ' no dispone de Correo.' from alumnos where email is null;</pre>		5.0

Figura 56: Resultados de las pruebas.

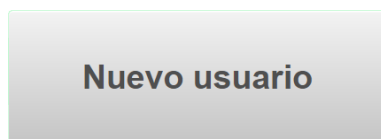
Anexo III

Manual del profesor

C. Manual del profesor

C.1. Crear un nuevo alumno

Para crear un nuevo alumno, tendrá que pulsar el botón



que le llevará a la vista (figura 57)

Una captura de pantalla de un formulario web con un fondo verde claro. El formulario está dividido en dos secciones: "Datos del usuario" y "Datos de matriculación". La sección "Datos del usuario" contiene campos para Expediente*, DNI/NIF*, Nombre*, Primer apellido*, Segundo apellido*, Teléfono*, E-mail*, Sexo* (menú desplegable), Dirección* y Fecha nacimiento* (con un icono de calendario). La sección "Datos de matriculación" contiene campos para Grupo*, Curso* y Asignatura* (menú desplegable). En la parte inferior del formulario hay dos botones: "Guardar" y "Cancelar".

Figura 57: Vista para crear un nuevo alumno.

en la cual tendrá que rellenar los campos:

■ Datos del alumno

- **Expediente:** Expediente del alumno (obligatorio).
- **DNI/NIF:** DNI o NIF del alumno (obligatorio).
- **Nombre:** Nombre del alumno (obligatorio).
- **Primer apellido:** Primer apellido del alumno (obligatorio).
- **Segundo apellido:** Segundo apellido del alumno.
- **Teléfono:** Teléfono del alumno.
- **E-mail:** Correo del alumno.
- **Sexo:** Sexo del alumno (masculino o femenino).
- **Dirección:** Dirección del alumno.
- **Fecha nacimiento:** Fecha de nacimiento del alumno.
- **Titulación:** Titulación que cursa el alumno.

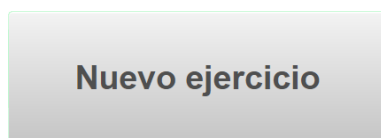
■ Datos de matriculación

- **Grupo:** Grupo que esta cursando (obligatorio).
- **Curso:** Año que esta cursando (obligatorio).
- **Asignatura:** Asignatura que esta cursando (obligatorio).

Finalmente pulsar el botón guardar. Si se produce algún error se avisa con un mensaje.

C.2. Crear un nuevo ejercicio

Para crear un nuevo ejercicio, tendrá que pulsar el botón



que le llevará a la vista (figura 58)

 A screenshot of a web form titled "Datos del ejercicio". The form includes several input fields: "Código: 49", "Enunciado*" (a large text area), "Etiquetas*" (a text input), "Esquema BD*" (a text input), "Esquema:" (a text input), "Dificultad*" (a text input), and "Tipo:" (a dropdown menu with "Selecciona tipo" as the selected option). Below these is a section titled "Añadir soluciones" which contains a "Solucion:" text area, a "Puntos:" text input, and an "Añadir" button. At the bottom of this section is a table with columns "Id", "Solución", and "Puntos", containing the text "Sin soluciones". At the very bottom of the form are two buttons: "Guardar" and "Cancelar".

Figura 58: Vista para crear un nuevo ejercicio.

en la cual tendrá que rellenar los campos:

- **Enunciado:** Enunciado del ejercicio (obligatorio).
- **Etiquetas:** Etiquetas que recibirá ese ejercicio (obligatorio). Estas etiquetas luego sirven para que en el modo entrenamiento se generen ejercicios dependiendo el nivel.
- **Esquema BD:** Esquema donde se corregirá el ejercicio (obligatorio).
- **Esquema:** Esquema que deberá seguir la sentencia introducida como respuesta del alumno.
- **Dificultad:** Dificultad del ejercicio (obligatorio). Al igual que las etiquetas, sirve para el modo entrenamiento.
- **Tipo:** Indica el tipo de ejercicio (obligatorio). Podemos encontrar ejercicios de evaluación, entrenamiento o reto.

Y añadir todas las soluciones propuestas para dicho enunciado, con sus correspondientes puntos. Tendrá que ir añadiéndolas a la lista.

- **Solución:** Solución o soluciones propuestas para la corrección.
- **Puntos:** Puntuación o puntuaciones que los alumnos obtendrán al responder los ejercicios una vez corregidos.

C.3. Administrar

C.3.1. Usuarios

Para administrar los usuarios, tendrá que pulsar el botón

Administración → Administrar usuarios

que le llevará a la vista, donde una vez seleccionado un alumno, veremos los diferentes apartados (figura 59)

Expediente	DNI/NIF	Nombre	Primer apellido	Segundo apellido	Grupo	Curso	Nota
3	11111111N	Alejandro	Fernandez	Miñambres	A	2016	


 <input type="button" value="Borrar avatar"/>	Información Titulación: <input type="text" value="Ing.Informatica"/> Dirección: <input type="text" value="Malaga"/> E-mail: <input type="text" value="ale1@ejemplo.com"/> Teléfono: <input type="text"/> Perfil de usuario Alias: <input type="text" value="3"/> Fecha nacimiento: <input type="text" value="17/03/2004"/> Sexo: <input type="text" value="Hombre"/> Fecha alta: <input type="text" value="19/05/1994"/> Nivel: <input type="text" value="0"/> Experiencia: <input type="text" value="0"/> <input type="button" value="Resetear Password"/> <input type="button" value="Aplicar cambios"/>
-------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 59: Vista para editar un usuario.


donde podemos:

- **Editar la información:** Editar la información y dale al botón “Aplicar cambios”.
- **Borrar el avatar:** Pulsar el botón “Borrar avatar”.
- **Resetear la contraseña:** Cambia la contraseña y la pone igual que el usuario. Pulsar el botón “Resetear password”.

C.3.2. Preguntas

Para administrar las preguntas, tendrá que pulsar los botones

Administración → Administrar preguntas

que le llevará a la vista, en la cual una vez seleccionado una pregunta , veremos los diferentes apartados que podremos editar (figura 60).

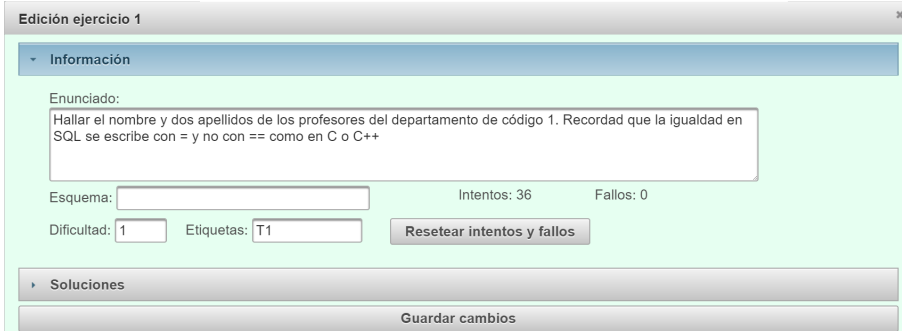


Figura 60: Vista para editar una pregunta.

Una vez terminado, pulsar en el botón “Guardar cambios”.

C.4. Ver informes de errores

Para ver los informes de errores que han mandado los diferentes alumnos, tendrá que pulsar el botón

Informes de error

que le llevará a la vista, donde verá una lista (figura 61).




Informes de error					
	Ej	Expediente	Descripción	Fecha	
0	9	10	Hola, tenemos un error aqui.	06/06/17	 

Figura 61: Lista de informes.

Si pulsa sobre , accederá al ejercicio en cuestión, podrá editarlo y finalmente podrá cerrar el informe.

C.5. Crear una prueba

Para crear una prueba, tendrá que pulsar el botón

Crear una prueba

con el cual accederá a la vista (figura 62) en la cual tendrá que rellenar los campos y seleccionar los ejercicios de los que esta compuesta la prueba.

Título: Grupo: Fecha de inicio: 07/06/2017 19:34:52

Palabra de paso: Fecha de fin:

Ejercicios disponibles

	Enunciado	Etiqueta	Dificultad	Esquema BD	Tipo
+	Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++	T1	1	BD	
+	Hallar el nombre y dos apellidos de los profesores de todos los departamentos salvo el de código 3. Para el operador "distinto" se puede utilizar != o bien <>.	T1	1	BD	
+	Hallar el nombre y dos apellidos de los profesores cuyo correo está en el servidor "cc.uma.es". En este caso hay que utilizar el operador LIKE. Recordad que nombre LIKE 'A%' será verdadero si nombre empieza por 'A'.	T1	1	BD	
+	Mostrar la lista de alumnos que no disponen de correo electrónico. Recordad el uso del operador IS NULL. Probad poniendo la expresión WHERE email = NULL .	T2	1	BD	

Figura 62: Vista para crear una prueba.

Los campos son:

- **Título:** Título de la prueba (obligatorio).
- **Grupo:** Grupo al cual ira dirigida la prueba (obligatorio).
- **Fecha inicio y fecha fin:** Fecha de inicio y fin de la prueba (obligatorio).
- **Palabra de paso:** Palabra para acceder a la prueba (obligatorio).

C.6. Crear un reto

C.6.1. Validar retos

Para validar un reto, tendrá que pulsar los botones

→

Una vez allí, verá una lista de todos los retos enviados por los alumnoosh, donde pulsando el botón accederá al ejercicio. Podrá editarlo y tendrá que indicar la puntuación para el creador (figura 63) y la puntuación para los alumnos que solucionen dicho ejercicio (figura 64).

Puntos creador:

Figura 63: Puntos creador.

	Solución	Descripcion	Pts
1	select *from profesores		20

Figura 64: Puntos del reto.

C.6.2. Crear reto

Para crear un reto, tendrá que pulsar los botones

Crear reto → **Crear retos**

donde llegará a la vista (figura 65) en la cual tendrá que rellenar los campos y seleccionar los ejercicios que formaran el reto.

Creador	Enunciado	Etiqueta	Dificultad
USU10	Seleccionar todos los profesores.		0

Figura 65: Vista para crear un reto.

Los campos son:

- **Título:** Título del reto (obligatorio).
- **Grupo:** Grupo al que va dirigido el reto (obligatorio).

C.7. Ver resultados de los alumnos

Para acceder a esta vista tendrá que pulsar el botón


Ver resultados de los alumnos

donde podrá corregir pruebas, habilitar o deshabilitar su visibilidad y ver los resultados de los alumnos (figura 66).

Cod.	Nombre asignatura	Grupo	Prueba
112	Base de datos	A	PruebaEjemp

Figura 66: Vista para ver los resultados.



C.7.1. Corregir pruebas

En el apartado donde se encuentran las pruebas disponibles para corregir (figura 67), pulsar el botón  en la prueba que se desea corregir.

Pruebas disponibles para corregir				
Cod.	Nombre asignatura	Grupo ↕	Prueba	
112	Base de datos	A	PruebaEjemp	

Figura 67: Pruebas disponibles para corregir.

C.7.2. Habilitar o deshabilitar visibilidad de una prueba


En el apartado donde se encuentran las pruebas que ya han sido corregidas (figura 68), pulsar sobre el botón  para habilitar la visibilidad o en el botón  para deshabilitar la visibilidad.

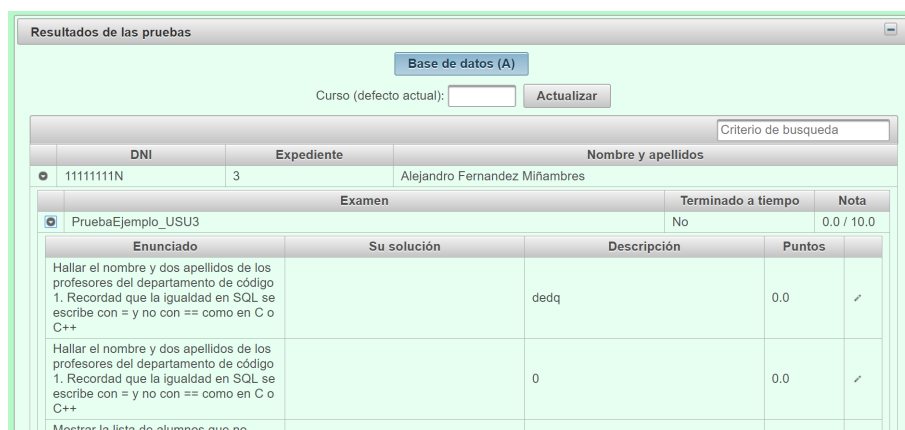
Pruebas disponibles para corregir				
Cod.	Nombre asignatura	Grupo ↕	Prueba	
112	Base de datos	A	PruebaEjemp	

Figura 68: Habilitar o deshabilitar la visibilidad.

C.7.3. Ver resultados de los alumno

En el apartado donde se encuentran los resultados de las pruebas (figura 69). Deberá pulsar sobre la asignatura que desea ver y también permite filtrar por curso (por defecto es el actual).

Una vez seleccionado un alumno y una prueba corregida, podrá ver los resultados y si desea modificarlos deberá pulsar sobre .

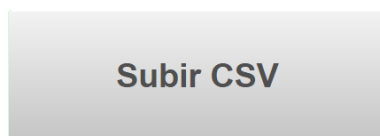


Resultados de las pruebas					
Base de datos (A)					
Curso (defecto actual): <input type="text"/> Actualizar					
Criterio de búsqueda					
DNI	Expediente	Nombre y apellidos			
11111111N	3	Alejandro Fernandez Miñambres			
Examen				Terminado a tiempo	Nota
PruebaEjemplo_USU3				No	0.0 / 10.0
Enunciado	Su solución	Descripción	Puntos		
Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++		dedq	0.0	✓	
Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++		0	0.0	✓	
Mostrar la lista de alumnos que no					

Figura 69: Resultados de las pruebas.

C.8. Subir alumno o ejercicios mediante un archivo CSV

Para subir alumnos o ejercicios mediante un archivo CSV, pulse el botón



y accederá a la vista (figura 70) donde mediante el botón **+ Choose** seleccionará el archivo a subir y con el botón **Upload** lo subirá al sistema.

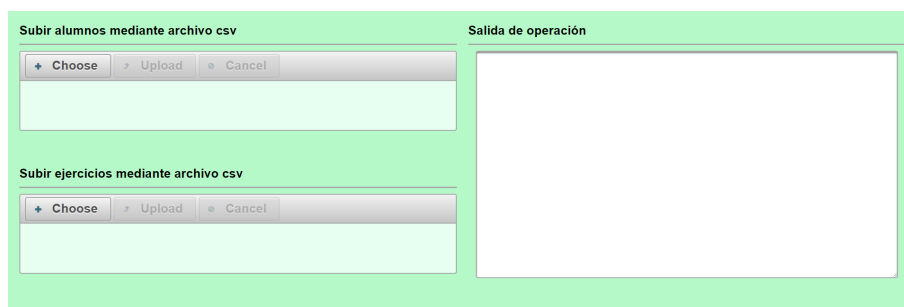


Figura 70: Subir mediante archivo CSV.

!Advertencia - CSV de alumnos;

El orden de las columnas es: **expediente, dni, nombre, primer apellido, segundo apellido, teléfono, sexo, dirección, email, fecha alta, fecha nacimiento, titulación, usuario, curso, grupo, asignatura matriculada.**

!Advertencia - CSV de ejercicios;

El orden de las columnas es: **enunciado, esquema, BD sobre el cual se corrige, tipo de ejercicio, etiquetas (separadas por comas), dificultad, solución, puntuación de la solución y descripción de la solución.**

Anexo IV

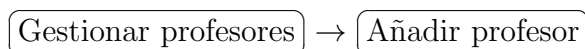
Manual del administrador

D. Manual del administrador

D.1. Gestión de profesores

D.1.1. Crear

Para crear un nuevo profesor y darlo de alta, tendrá que pulsar los botones



que le llevará a la vista (figura 71)

NRP*:	<input type="text"/>	Id usuario*:	<input type="text"/>
Nombre*:	<input type="text"/>	Primer apellido*:	<input type="text"/>
Segundo apellido:	<input type="text"/>	Sexo:	<input type="text"/>
E-mail:	<input type="text"/>	Departamento:	<input type="text"/>
Fecha nacimiento:	<input type="text"/>	Antigüedad:	<input type="text"/>
Contraseña*:	<input type="text"/>		

Guardar Cerrar

Figura 71: Crear profesor.

en la cual tendrá que rellenar los campos:

- **NRP:** Código con el que se identifica el profesor (obligatorio).
- **Id usuario:** Nombre para poder iniciar sesión en la aplicación (obligatorio). Debe empezar por una letra.
- **Nombre:** Nombre del profesor (obligatorio).
- **Primer apellido:** Primer apellido del profesor (obligatorio).
- **Segundo apellido:** Segundo apellido del profesor.
- **Sexo:** Sexo del profesor (masculino o femenino).
- **E-mail:** E-mail del profesor.
- **Departamento:** Departamento donde trabaja el profesor.
- **Fecha nacimiento:** Fecha de nacimiento del profesor.
- **Antigüedad:** Antigüedad del profesor.
- **Contraseña:** Contraseña para iniciar sesión en la aplicación (obligatorio).

Una vez rellenado al menos, todos los campos obligatorios, podrá guardar el profesor y se le dará de alta en el sistema. En caso de que se produzca algún error, se avisará con un mensaje.

D.1.2. Editar y asignar asignaturas que imparte

Para editar la información de un profesor, tendrá que pulsar el botón



donde accederá a la vista (figura 72)

Editar profesor 1

NRP: 1 Id usuario: PROFESOR
Nombre: Alejandro Primer apellido: Fernández
Segundo apellido: Sexo: Masculino
E-mail: ale@ejemplo.com Departamento: 2
Fecha nacimiento: Antigüedad:

Imparte			
	Truno	Código asignatura	Curso
	A	112	2016

Truno:
Codigo asignatura:
Curso:

Figura 72: Editar y asignar una asignatura.

en la cual podrá editar todos los campos mencionados en la sección para crear un profesor (sección 8) y además podrás asignarle las asignaturas que imparte.

Para ello tendrá que rellenar los campos, turno, código asignatura (asegurarse de que existe dicha asignatura), curso y pulsar añadir.

D.1.3. Borrar

Para borrar un profesor, tendrá que pulsar el botón

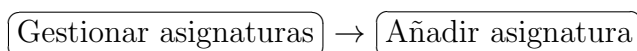


y el profesor será borrado y dado de baja en el sistema.

D.2. Gestión de asignaturas

D.2.1. Crear

Para crear una nueva asignatura, tiene que pulsar



mostrándose la vista (figura 73)

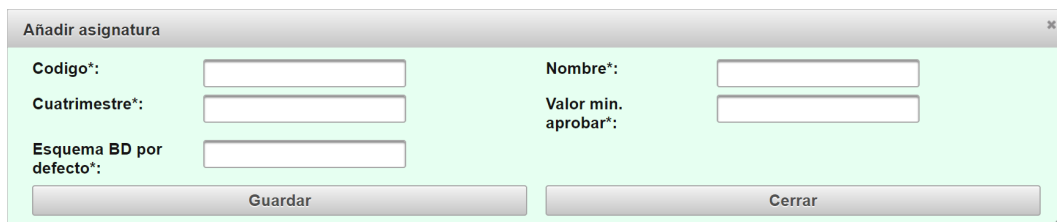


Figura 73: Crear asignatura.

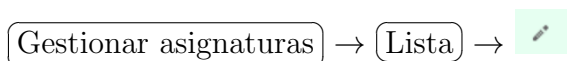
en la cual tendrá que rellenar los campos:

- **Código:** Código con el cual se identifica la asignatura (obligatorio).
- **Nombre:** Nombre de la asignatura (obligatorio).
- **Cuatrimestre:** Cuatrimestre en el cual se imparte la asignatura (obligatorio).
- **Valor mínimo para aprobar:** Valor mínimo para dar la asignatura como aprobada (obligatorio).
- **Esquema BD por defecto:** Indica la base de datos sobre la cual serán corregido los retos (obligatorio).

Una vez rellenado los campos, pulsar guardar y la asignatura será creada. En caso de que se produzca algún error, se avisa con un mensaje.

D.2.2. Editar

Para editar la información de una asignatura, tendrá que pulsar



donde se activará el modo para editar la asignatura (figura 74)

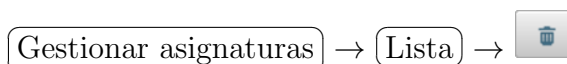


Figura 74: Editar asignatura.

en la cual podrá editar los diferentes campos.

D.2.3. Borrar

Para borrar una asignatura, tendrá que pulsar



y la asignatura será borrada.

Importante

Antes de borrar una asignatura, debe tener en cuenta que ningún profesor la imparta y que ningún alumno la curse. En caso de intentarlo, no podrá ser borrada.

D.3. Gestión de procesos

D.3.1. Crear

Para crear un nuevo proceso, tiene pulsar los botones

Gestionar procesos → Añadir proceso

mostrándose la vista (figura 75)

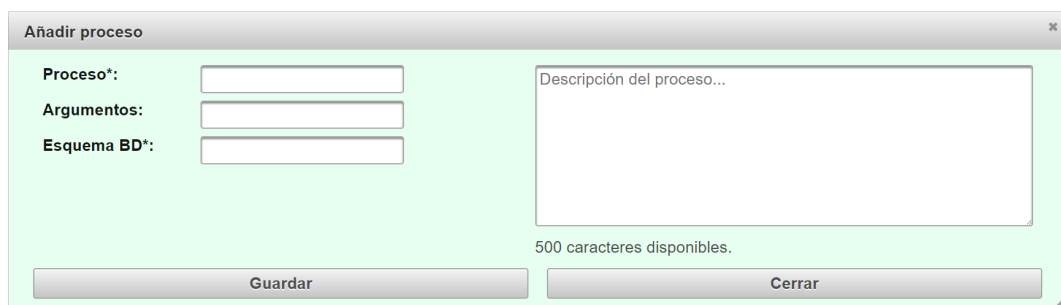


Figura 75: Crear proceso.

en la cual tendrá que rellenar los campos:

- **Proceso:** Nombre con el cual identificamos el proceso (obligatorio).
- **Argumentos:** Argumentos que necesite el proceso. Es necesario ponerlo tal cual se haría en una llamada a una función SQL. Ej: ('Alejandro')
- **Esquema BD:** Nombre del esquema sobre el cual se ejecutará el proceso (obligatorio).

Una vez rellenado los campos, pulsar guardar.

D.3.2. Editar

Para editar la información de un proceso, tendrá que pulsar

Gestionar proceso → Lista

y pulsar encima del campo a editar (76). Solo se permite editar los campos descripción y argumentos.

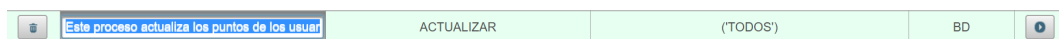


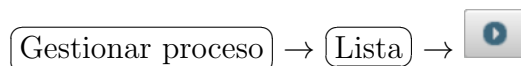
Figura 76: Editar proceso.

D.3.3. Ejecutar

Importante

Para poder ejecutar un proceso, este debe existir previamente en la base de datos donde deseamos ejecutarlo. En caso de no existir no se ejecuta.

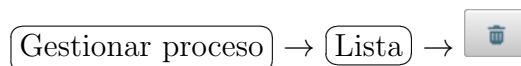
Para ejecutar un proceso, tendrá pulsar



donde una vez pulsado el proceso será ejecutado.

D.3.4. Borrar

Para borrar un proceso, tendrá que pulsar



y el proceso será borrado.

Anexo V

Manual del instalación

E. Manual de instalación

La instalación de la aplicación es bastante sencilla y su explicación se hará mediante un ejemplo.

Advertencia

El procedimiento de instalación se basa en un ejemplo, por lo tanto a la hora de instalarlo usted puede cambiar los parámetros acorde con la configuración deseada.

En este ejemplo, la instalación se realizó sobre **Apache Tomcat 8.0.27** y **Oracle Database 11g Express Edition Release 11.2.0.2.0**

E.1. Preparación de la base de datos

Una vez iniciada la sesión como **SYS** en nuestra base de datos.

E.1.1. Paso 1: Creación del tablespace

```
CREATE TABLESPACE TS_SQLGAME DATAFILE 'TS_SQLGAME.DAT' SIZE 200M AUTOEXTEND
ON;
```

E.1.2. Paso 2: Creación del usuarios que contendrá la aplicación

```
CREATE USER GAMESQL IDENTIFIED BY BD DEFAULT TABLESPACE TS_SQLGAME
QUOTA UNLIMITED ON TS_SQLGAME;
```

```
GRANT ALL PRIVILEGES TO GAMESQL;
```

E.1.3. Paso 3: Creación de los roles con sus privilegios

```
CREATE ROLE R_PROFESOR;
GRANT CREATE SESSION TO R_ADMINISTRADOR;
```

```
REATE ROLE R_ADMINISTRADOR;
GRANT CREATE SESSION TO R_PROFESOR;
```

```
CREATE ROLE R_ALUMNO;
GRANT CREATE SESSION TO R_ALUMNO;
```

E.1.4. Paso 4: Creación del esquema GameSQL

Siendo el usuario **GAMESQL** ejecutar el script que contiene el modelo entidad-relación, los procedimientos, etc (*GameSQL - Base de datos.ddl*).

E.1.5. Paso 5: Creación de usuario y esquema de las bases de datos externas

Crear las bases de datos sobre las cuales se ejecutará las sentencias a corregir y añadir a estas bases de datos los procedimientos externos incluidos en el script (*GameSQL - Procedimientos externos.ddl*).

E.1.6. Paso 6: Creación de un usuario administrador

```
CREATE USER ADMIN IDENTIFIED BY BD;  
GRANT R_ADMINISTRADOR TO ADMIN;
```

E.2. Preparación de la aplicación

Dentro del WAR generado, acceder a:

```
$ ./WEB-INF/classes/configuracion.properties
```

Este archivo contiene la configuración que permite el enlace con la base de datos y tendremos que editarlo según nuestras necesidades.

- **ip_bd:** Ip de la base de datos.
- **port:** Puerto de la base de datos.
- **nombre:** Nombre de la base de datos (SID).
- **userBD:** Usuario de la base de datos que contiene el modelo y procedimientos de nuestra aplicación.
- **pwdBD:** Contraseña de userBD.

Desplegar la aplicación y arrancarla.