

# QueryCompetition: un sistema web para practicar consultas en SQL

Erick Orlando Matla Cruz<sup>1</sup>, Miguel Ehécatl Morales Trujillo<sup>2</sup> y David Velázquez Portilla<sup>3</sup>

<sup>1</sup>Posgrado en Ciencias de la Administración

<sup>2</sup>Facultad de Ingeniería

<sup>3</sup>Coordinación de Cómputo e Informática del Posgrado de la Facultad de Medicina

Universidad Nacional Autónoma de México

Ciudad Universitaria, México

{ematla, migmor, davepo}@ciencias.unam.mx

## Resumen

En este artículo se presenta QueryCompetition, un sistema web ideado para cursos de Bases de Datos, que permite al profesor crear concursos de resolución de consultas en SQL cuyos participantes son los alumnos de su clase. QueryCompetition se creó buscando que los alumnos practiquen lo aprendido, que se motiven a mejorar sus habilidades en un ambiente competitivo sano, y que sean más precisos y conscientes al responder las consultas. QueryCompetition demostró ser una herramienta útil y práctica que motiva a los alumnos y ayuda al profesor a medir el aprendizaje.

## Abstract

This paper presents QueryCompetition, a web system designed for Databases courses. It allows the teacher to create quiz contests in SQL tailored for students of his/her class. QueryCompetition was created seeking that students could practice what they have learned, to motivate them to boost their knowledge in a healthy competitive environment, and to become more precise and aware when answering queries. It was found that QueryCompetition is a useful and practical tool that motivates students and helps the teacher to measure learning.

## Palabras clave

Sistema web, concurso, competición, SQL, bases de datos.

## 1. Introducción

La búsqueda de alternativas pedagógicas que motiven a los alumnos a aprender es una labor que ocupa a

los profesores en cualquier nivel educativo. En el caso de la enseñanza de asignaturas donde la tecnología es una constante, se debe aprovechar que los alumnos ya se encuentran en contacto con ésta para utilizarla en el proceso de aprendizaje.

Este es el caso de asignaturas relacionadas con las bases de datos, donde el uso de herramientas informáticas es inevitable e indispensable. Bases de Datos es una asignatura impartida de forma obligatoria para alumnos de la carrera de Ciencias de la Computación y de manera optativa para las carreras de Actuaría y Matemáticas de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM). La demanda de esta asignatura en estas carreras se ha incrementado de manera considerable. En parte, el incremento se debe a que los egresados de estas carreras en el campo laboral tienen contacto, en distintos niveles, con bases de datos. Por esta razón, en cualquier curso resulta importante que el alumno practique lo aprendido, en el caso particular de los conocimientos y habilidades relacionados con bases de datos: el uso de sistemas de gestión, sistemas web y por supuesto SQL.

Para ello es necesario conjuntar la teoría mostrada en clase con la experiencia práctica que permita al alumno aplicar y reforzar los conocimientos adquiridos. Sin embargo se deben tomar en cuenta varias restricciones que recaen sobre los cursos, como lo son el tiempo, los temas a cubrir, el perfil deseado del egresado, su conocimiento previo y la motivación de los alumnos por aprender. De acuerdo con [9], para aprender algo es necesario contar con las capacidades, conocimientos y destrezas necesarias (poder), además de tener la disposición e intención suficientes para alcanzar determinadas metas (querer). El poder no implica el querer, por lo que es necesario otorgarle a las metas un valor que sea consistente con los deseos de los alumnos. En [2] se identifican cuatro tipos de valores como factores que determinan cómo se realiza una tarea: el valor de *logro*, *intrínseco*, *utilidad* y *coste*. Para

finés de este trabajo resultan de interés dos de ellos: el valor de logro y el valor intrínseco. El valor del logro está dado por la importancia que el estudiante da a “realizar bien una tarea”, mientras que el valor intrínseco está definido como la satisfacción obtenida de su actuación en el desarrollo de una actividad [2].

Para conjugar aspectos motivacionales durante la enseñanza de bases de datos, este trabajo presenta QueryCompetition, un sistema web que permite a los alumnos practicar sus conocimientos de SQL para resolver consultas. Por una parte, el sistema tiene como objetivo proveer a los alumnos de una herramienta para practicar lo aprendido y con ello saber “qué tan bien hacen una tarea” (valor del logro) en un ambiente competitivo donde hay un reconocimiento público a los mejores (valor intrínseco).

Por otra parte, el sistema permite al profesor estructurar una competencia entre los alumnos en donde se pondere la eficiencia, velocidad de respuesta y, por supuesto, el uso correcto de SQL, lo que permite, de manera colateral, evaluar el aprendizaje de sus alumnos.

Este artículo se compone de las siguientes partes: en la sección 2 se presentan los antecedentes de sistemas relacionados con resolución de consultas y aprendizaje de SQL. La sección 3 describe a QueryCompetition. La sección 4 muestra los resultados obtenidos, mientras que en la sección 5 se concentran las conclusiones y el trabajo futuro.

## 2. Antecedentes

La virtud está en encontrar el equilibrio entre el juego y la seriedad: el juego mantendrá interesados a nuestros estudiantes y motivará su discusión más allá de las paredes del aula [4]. Teniendo en cuenta esta afirmación, resulta prometedor para los docentes ir en busca de ese equilibrio.

En el campo de la informática, existen propuestas interesantes que involucran el uso de competencias como medio de aprendizaje. En [1] se reporta una experiencia docente positiva basada en una competencia entre los alumnos como parte de la asignatura. La competencia consistió en que cada alumno debía implementar un algoritmo para el juego Oteho, posteriormente, los algoritmos desarrollados se enfrentarían en partidas. Los resultados obtenidos al término de la competencia fueron: una mayor motivación de los alumnos, un incremento en la participación e involucramiento en las actividades de clase y una mejora en las calificaciones obtenidas en la asignatura.

En [6] se describe una actividad alternativa para la evaluación del aprendizaje basada en el uso de una competencia entre pares. En esta competencia, cada equipo realizó un resumen, una presentación oral y un debate sobre algún tema de la asignatura establecido

por el profesor. Posteriormente, se desarrolló una liga *todos contra todos* en la que cada equipo evaluó el nivel de conocimiento que cada equipo había alcanzado sobre el tema que ellos mismos expusieron. Entre los resultados de esta experiencia destacan un incremento en la satisfacción del alumno, un aumento en la motivación y, sobre todo, una percepción del alumno de haber profundizado más en el tema expuesto en comparación con los mecanismos de aprendizaje tradicionales.

Es notorio que las competencias incrementan la motivación de los alumnos y proporcionan un mecanismo para que el alumno demuestre sus habilidades, sus proyectos, comparta sus intereses y pueda compararse con otros [8]. De acuerdo con [3] el uso de juegos serios incrementa significativamente el aprendizaje en el área de la informática.

Por otra parte, las competencias de programación son un estímulo importante para los alumnos de informática [5]. En particular para programación, en la red existen alternativas para aprender y practicar SQL. Entre ellas se pueden mencionar: HackerRank<sup>1</sup>, SQLZoo<sup>2</sup>, PL/SQL Challenge<sup>3</sup> o HackerEarth<sup>4</sup>.

HackerRank es una plataforma en línea que presenta una consulta, las tablas necesarias para responder la consulta y la consola para que el usuario introduzca el código SQL. Una ventaja de HackerRank es que el usuario puede elegir entre cuatro sistemas manejadores. Además, es posible seleccionar el editor de texto y el tema. Cuenta con 4 pestañas: en la que se presenta el problema a resolver, las respuestas que has enviado, la tabla de posiciones de los usuarios y un foro.

SQLZoo, además de presentar problemas de resolución a través de SQL, también cuenta con cuestionarios para probar el conocimiento teórico del usuario y acceso directo a documentación de SQL. Los problemas presentados por SQLZoo están basados en modificar fragmentos de código dados.

PL/SQL Challenge presenta al usuario retos y cuestionarios que deben ser resueltos mediante código SQL o PL/SQL principalmente. La desventaja de esta plataforma es que tanto los retos como los cuestionarios son exclusivos de Oracle.

HackerEarth, al enviar la respuesta presenta un informe más completo que SQLZoo o HackerRank, ya que incluye elementos como el tiempo de ejecución y la puntuación obtenida.

Después de revisar las plataformas anteriormente mencionadas, se identificaron algunas oportunidades de mejora sobre las cuales QueryCompetition podría trabajar. Entre estas mejoras destacan:

<sup>1</sup>HackerRank, [www.hackerrank.com](http://www.hackerrank.com)

<sup>2</sup>SQLZoo, <http://sqlzoo.net>

<sup>3</sup>PL/SQL Challenge, <https://plsqlchallenge.oracle.com>

<sup>4</sup>HackerEarth, [www.hackerearth.com](http://www.hackerearth.com)

- Permitir al usuario definir consultas propias.
- Permitir al usuario decidir sobre qué base de datos se trabajará.
- Establecer quiénes pueden participar en la resolución de los problemas.
- Crear grupos para competir con un conjunto determinado de personas.

Una diferencia importante entre las plataformas existentes y el objetivo que persigue QueryCompetition es que las primeras tienen un mayor peso sobre el aprendizaje autodidacta, mientras que QueryCompetition busca que los alumnos, inmersos en un ambiente competitivo, se motiven a practicar más y en consecuencia mejorar su aprendizaje.

### 3. QueryCompetition

En esta sección se presenta la descripción del sistema, el mecanismo de evaluación automático de respuestas y las especificaciones técnicas de QueryCompetition.

#### 3.1. Descripción del sistema

QueryCompetition<sup>5</sup> es un sistema web ideado para practicar consultas en SQL. Este sistema permite al profesor crear concursos de resolución de consultas en SQL cuyos participantes son los alumnos de su clase.

Los concursos creados en QueryCompetition están compuestos por rondas en las que es posible configurar aspectos como: la hora de inicio y cierre de cada una, indicar qué alumnos participarán en una ronda determinada, quiénes clasificarán a una siguiente ronda, cargar las consultas que se utilizarán como reactivos, establecer la base de datos sobre la que se ejecutarán las consultas, definir los puntos otorgados por cada consulta resuelta correctamente, por la eficiencia de ésta y por la velocidad de respuesta, entre otras funciones.

Una vez que el profesor se registre en el sistema se le asignará un usuario y contraseña, a partir de ese momento, el flujo básico de uso de QueryCompetition es:

1. El profesor da de alta al grupo, la universidad a la que éste pertenece y el semestre en curso.
2. El profesor registra a los alumnos que participarán en el concurso, asignándoles un usuario y contraseña.
3. El profesor establece los datos de conexión a la base de datos sobre la que se ejecutarán las consultas, esto es: la dirección IP, el puerto, el nombre de la base de datos, el usuario y la contraseña.

<sup>5</sup>Accesible desde <http://132.247.127.131:8080/QueryCompetition/> con el usuario "jenui#", donde # puede ser substituido por {00, 01, 02, ..., 99}, y contraseña "J3nu1" para todos los usuarios.

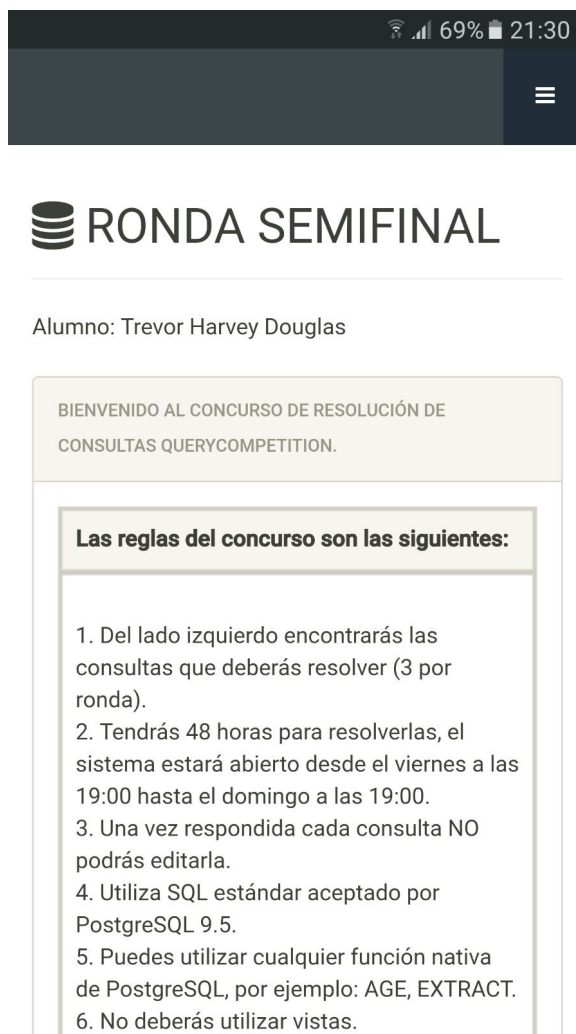


Figura 1: Pantalla inicial de ronda (versión móvil).

4. El profesor carga las consultas en lenguaje natural que servirán como reactivos así como la consulta en SQL que arroja el resultado esperado.
5. El profesor define las rondas que conformarán el concurso, asociando los alumnos que participarán y las consultas que serán presentadas en cada ronda a los alumnos.
6. Llegado el momento, el sistema abre la ronda para que los alumnos puedan responder las consultas (ver Figura 1 y 2). Al cumplirse el plazo establecido el sistema cierra la ronda.
7. El profesor solicita la calificación de la ronda al sistema y vuelve a 5 (ver Figura 3).

El sistema realiza la evaluación automática de las respuestas y cuenta con vistas para que el profesor analice los resultados de los alumnos por ronda y/o concurso, de manera individual y/o grupal, ya sea por puntos obtenidos y/o respuestas correctas.

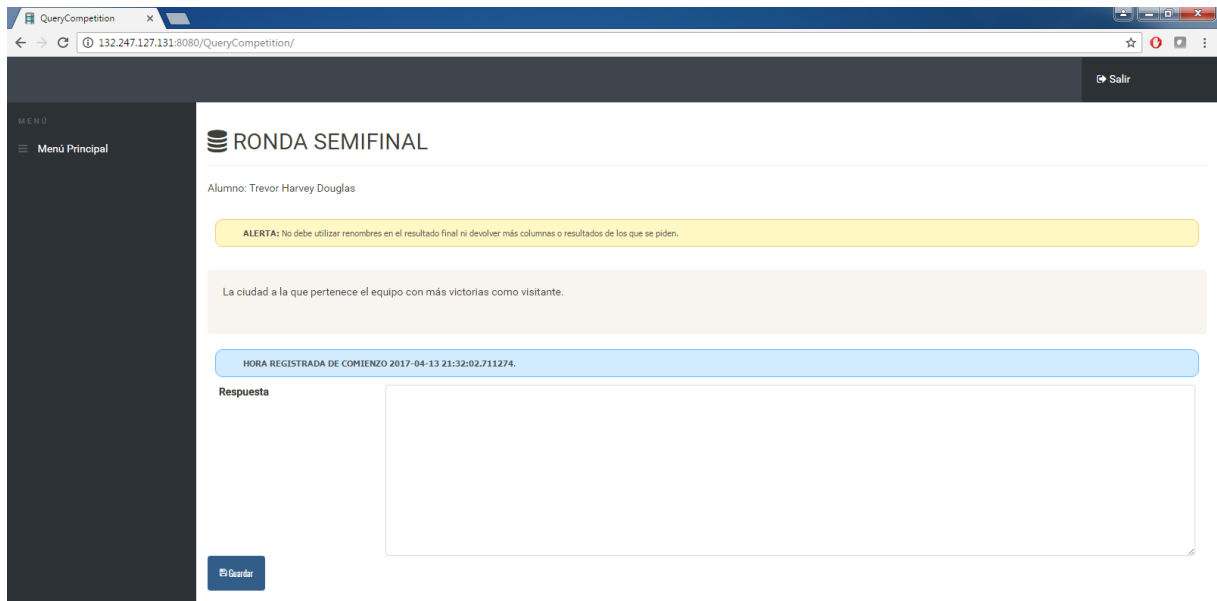


Figura 2: Consulta mostrada al alumno (versión web).

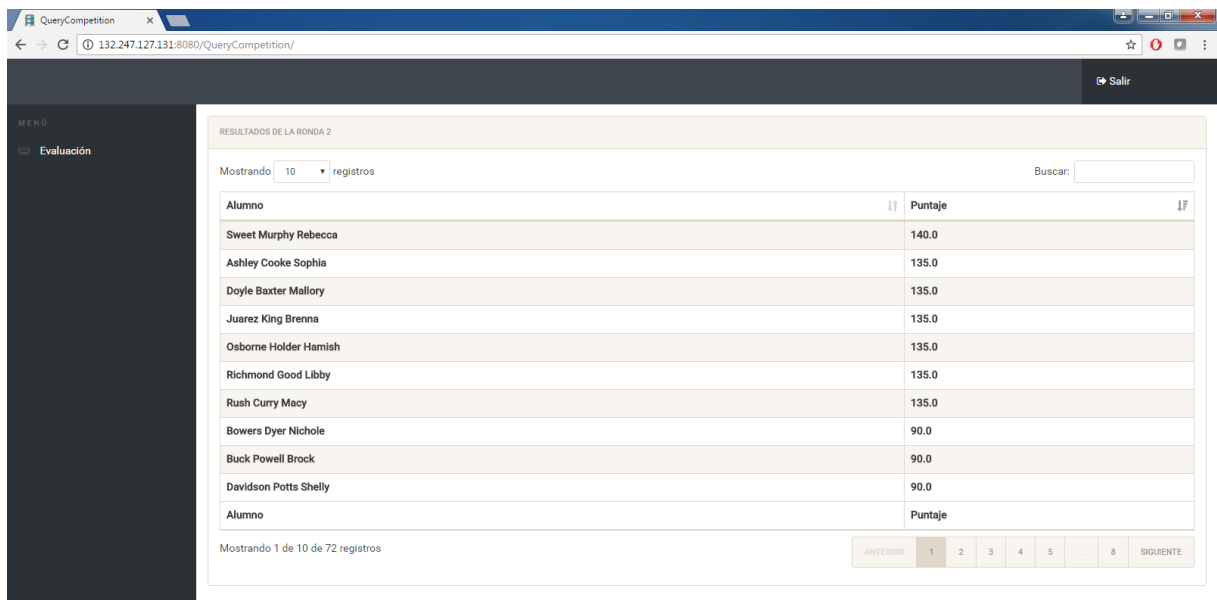


Figura 3: Pantalla con los resultados de la ronda (versión web).

### 3.2. Evaluación automática de respuestas

QueryCompetition establece algunas restricciones sobre las respuestas para que éstas puedan ser evaluadas automáticamente. Ejemplos de ellas son:

1. Una vez enviada la respuesta, ésta no puede editarse.
2. Se deberá utilizar SQL estándar aceptado por PostgreSQL.
3. Está permitido utilizar cualquier función nativa de PostgreSQL, por ejemplo: AGE o EXTRACT.
4. No se deberá usar renombramiento (AS) sobre la consulta más externa de la respuesta.
5. No está permitido utilizar vistas ni funciones definidas por el usuario.

QueryCompetition evalúa las consultas de manera automática, esto se logra comparando el resultado (*result set*) que regresa la consulta enviada por el alumno contra el resultado de la consulta correcta que el profesor proveyó al crear el concurso.

Determinar que dos consultas son equivalentes es un problema sumamente complejo. De acuerdo con Rice "dada una propiedad no trivial de las funciones parciales, no es computable determinar si una función arbitraria la posee o no"[10]. Dicho de otra manera, determinar si dos funciones son equivalentes es un problema computacional indecidible [7]. Trasladado al contexto de QueryCompetition, no es posible construir un programa que decida, para todos los casos, si dos consultas en SQL son equivalentes. Nótese que se está hablando de consultas no triviales, es decir, que difieren sintácticamente una de otra.

Más aún, que dos consultas arrojen el mismo resultado para todos los casos implica que ambas sean semánticamente equivalentes. Por lo que al ser un problema semántico, se descartan otros métodos de evaluación sintácticos como podría ser comparar los planes de ejecución de las consultas. Entre la comunidad de desarrolladores de bases de datos y programadores, la opción elegida para resolver este problema se centra en la comparación de los resultados de las consultas. Dicha comparación se lleva a cabo a través de comandos nativos de SQL como EXCEPT o el uso de cursores para recorrer ambos resultados, entre otras alternativas.

En QueryCompetition, la comparación de las consultas se realizó mediante una función que recibe dos cadenas:  $Qp$  y  $Qa$ . Donde  $Qp$  es la cadena que contiene a la consulta proveída por el profesor y  $Qa$  es la cadena con la consulta enviada por el alumno. De manera general el proceso de evaluación de la respuesta del alumno es el siguiente:

1. Limpiar  $Qa$  de caracteres especiales, punto y comas o comentarios.

2. Determinar qué tipo de resultado se espera a partir de  $Qp$ . El tipo de resultado pudiera ser una tabla o una función de agregación, estos casos deben manejarse de manera diferente, ya que el primero es un conjunto de tuplas y el segundo un valor.

En caso de que el tipo de resultado esperado sea una tabla:

1. Realizar un FULL OUTER JOIN entre  $Qp$  y  $Qa$ . De esta manera aquellas tuplas que arroja  $Qp$  pero no  $Qa$  tendrán un NULL en las columnas de la derecha. Análogamente, las tuplas que arroja  $Qa$  pero no  $Qp$  tendrán NULL en las columnas de la izquierda.
2. Contar cuántas tuplas tienen NULL en el lado izquierdo y cuántas en el derecho.
3. Verificar el valor de los dos contadores, si ambos son 0 la consulta se considera correcta, en otro caso incorrecta.

En caso de que el tipo de resultado esperado sea una función de agregación:

1. Comparar el valor arrojado por  $Qp$  contra el enviado por  $Qa$ , si son iguales la consulta se considera correcta, en otro caso incorrecta.

Para ambos casos, si  $Qa$  arroja un error de ejecución, automáticamente la respuesta se considera como incorrecta.

Cabe señalar que la función que evalúa si una consulta es correcta o no fue probada con 413 consultas de prueba, obteniendo un 100 % de efectividad. Sin embargo, esta función es susceptible a ser mejorada, para ello es necesario obtener retroalimentación. Esto se pretende lograr al socializar QueryCompetition entre profesores, alumnos y universidades con el fin de incrementar el número de usuarios y concursos llevados a cabo con este sistema.

Además de la función de comparación, se construyeron otras funciones que se ejecutaban en distintos momentos y para diferentes fines. Por ejemplo, cuando el alumno comienza el ejercicio (se obtiene la hora de inicio); cuando envía su respuesta (se determina la hora de término y se evalúa la respuesta); y cuando se cierra la ronda (se calculan los promedios de eficiencia y tiempo de respuesta).

Los promedios de eficiencia y tiempo de respuesta son calculados una vez que se cierra cada ronda. Ambos promedios se calculan para cada consulta que se usó como reactivo en la ronda. Posteriormente, mediante otra función, se compara cada promedio con la eficiencia y tiempo de respuesta del alumno. Es aquí cuando se otorgan los puntos asociados a cada métrica.

Finalmente, existen otras funciones auxiliares utilizadas en QueryCompetition, por ejemplo, para limpiar la respuesta ya que en ocasiones el alumno enviaba co-

mentarios "-" o incluía el ";", o para dar formato a la información que se despliega al mostrar las clasificaciones.

### 3.3. Especificaciones técnicas

Para la construcción de QueryCompetition se utilizó software libre: Java como lenguaje de programación, GlassFish como servidor de aplicaciones y PostgreSQL como sistema de gestión de bases de datos.

La arquitectura seguida fue Modelo-Vista-Controlador y el diseño de la interfaz de usuario es adaptativo para que se pueda utilizar desde cualquier dispositivo al que el alumno tenga acceso.

El diseño de la base de datos de QueryCompetition permite que varios profesores puedan hacer uso de la plataforma al mismo tiempo, así como tener múltiples concursos activos.

Si algún profesor, alumno o universidad estuviera interesada en utilizar QueryCompetition puede ponerse en contacto con los autores para proveerles una cuenta de acceso y configurar el concurso. Cuando se concluya con la nueva versión del sistema, se pondrá a disposición de los interesados el archivo WAR y el script de creación de base de datos. De esta manera bastará con tener un servidor con GlassFish y PostgreSQL instalados para poderlo utilizar de manera independiente.

## 4. Resultados

QueryCompetition se encuentra en su primera versión y fue probado en un grupo piloto con 82 alumnos y 2 profesores. El grupo estuvo compuesto por alumnos de séptimo y octavo semestres de diversas carreras físico-matemáticas de la Facultad de Ciencias de la Universidad Nacional Autónoma de México.

La base de datos utilizada para el concurso modela una liga de fútbol y está compuesta por 17 tablas, entre las que destacan: jugador, equipo, entrenador, ciudad y partido. Al comienzo del evento se distribuyó entre el grupo el script de creación de la base de datos, tanto esquema como datos, y el diagrama de tablas de dicha base de datos. Lo anterior se hizo con la intención de que los alumnos pudieran replicar el ambiente localmente y practicar de manera independiente.

El concurso se realizó en tres etapas: la primera consistió en 3 rondas clasificatorias en las que participaron remotamente los 82 alumnos. Cada ronda presentaba 3 consultas en lenguaje natural sobre la base de datos y contaban con 48 horas para responderlas. En particular, estas rondas comenzaban los días viernes a las 18:00 y concluían los domingos.

Al término de cada ronda clasificatoria, se publicaba la tabla de posiciones. Para realizar dicha clasificación

se tomaron en cuenta los siguientes criterios:

- Por cada consulta correcta, el alumno recibía 25 puntos.
- Si la consulta era más eficiente que el promedio de eficiencia del grupo, se asignaban 10 puntos adicionales.
- Si la consulta se respondía en menos tiempo que el promedio de tiempo que le tomó al grupo, se asignaban 5 puntos adicionales.

En consecuencia, un alumno podía obtener hasta 40 puntos por consulta correcta. Como se mencionó en el apartado anterior, los puntos asignados por criterio pueden ser ajustados por el profesor, ya que QueryCompetition provee las funcionalidades necesarias para configurarlos. La segunda etapa consistió en una ronda semifinal en la que únicamente participaron los 16 alumnos mejor colocados en la tabla general. Esta ronda se realizó de la misma manera que las rondas clasificatorias, es decir, remotamente y a 3 consultas.

La tercera etapa consistió en una eliminatoria presencial desarrollada en el aula. Esta etapa se dividió en 2 rondas de 1 consulta cada una y 10 minutos para resolverla. En la primera eliminatoria participaron los 8 alumnos mejor colocados en la tabla general. Para la segunda eliminatoria calificaron los 4 alumnos con mejor puntuación en la ronda previa. El campeón resultó ser el alumno que respondió correctamente la consulta en el menor tiempo posible.

En el cuadro 1 se resumen las características de cada una de las rondas del concurso.

Al concluir el primer concurso, se obtuvieron resultados satisfactorios reflejados en un incremento en la participación, una sana competencia, la integración del grupo y el reconocimiento al ganador. Es de resaltar que a la ronda final asistieron más de 60 alumnos como espectadores.

### 4.1. Mejoras propuestas

Entre las mejoras propuestas a la primera versión de QueryCompetition se identificaron las siguientes:

- Que el profesor pueda cargar automáticamente la lista de alumnos mediante un archivo CSV.
- Que las contraseñas se generen automáticamente.
- Que se envíen automáticamente avisos y alertas a través de correo electrónico.
- Que al término de cada etapa el alumno pueda consultar la respuesta esperada y compararla contra la que envió. A través de esta retroalimentación el alumno realizará un ejercicio de reflexión.
- Incluir el diagrama de tablas de la base de datos sobre la que se está trabajando en la vista del alumno.

Etapa	Ronda	Modalidad	Participantes	Consultas a resolver	Tiempo
1	1	Remota	82	3	48 horas
	2	Remota	82	3	48 horas
	3	Remota	82	3	48 horas
2	4	Remota	16	3	48 horas
3	5	Presencial	8	1	10 minutos
	6	Presencial	4	1	10 minutos

Cuadro 1: Estructura del concurso.

A través de una encuesta en línea que fue respondida por 69 de los 82 alumnos participantes en el concurso, se recolectaron las opiniones de los alumnos acerca del curso en general. Entre las opiniones recolectadas se incluyeron comentarios acerca de QueryCompetition. Entre las opiniones positivas de los alumnos acerca de esta primera edición del concurso destacan:

- “*Me pareció una muy buena dinámica Query Competition para poder practicar las consultas a una base*”.
- “*Query Competition me ayudó a reforzar el aprendizaje*”.
- “*Lo de la competencia fue una idea genial, aunque a lo mejor si fuera un poco antes de que terminara el curso sería mucho mejor*”.
- “*Promovió una sana competencia*”.

Sin embargo, no todas las opiniones fueron positivas, aquellas negativas pueden ser clasificadas en los siguientes aspectos:

- Dudar sobre si las consultas fueron calificadas correctamente.
- No saber manejar la presión por parte de algunos alumnos.
- Considerar injusto el hecho de que un error de sintaxis causara que la consulta fuera incorrecta.
- Considerar demasiado estricto que si el resultado de una consulta difería en “una” tupla, la consulta fuera tomada como incorrecta.

Para mitigar algunos de los puntos anteriores, se realizaron las siguientes acciones: (i) se publicaron las respuestas esperadas para que los alumnos las compararan contra las suyas. Esto se realizó en las rondas 1 a 3; y (ii) discutir y puntualizar en clase lo que implica un error de sintaxis y la diferencia entre que una consulta “regrese una tupla o no regrese ninguna”.

## 5. Conclusiones

El objetivo de este trabajo es ofrecer a los alumnos que cursan asignaturas relacionadas con bases de datos una herramienta que les permita poner en práctica, sin

la dirección estricta de un profesor, el conocimiento que adquieren.

La principal diferencia de QueryCompetition con respecto a sistemas similares radica en que QueryCompetition aprovecha el factor competitivo para motivar a los alumnos a mejorar sus habilidades y el cómo aplican lo aprendido. Mientras que otros sistemas encaminan sus objetivos a que el estudiante aprenda y se auto-evalúe, QueryCompetition se enfoca en la explotación del conocimiento adquirido previamente, aprovechando el espíritu competitivo del alumno.

El aspecto motivacional en el que se fundamentó QueryCompetition fue el de incrementar el valor del logro y el valor intrínseco en los alumnos. Por una parte, la ambición individual del alumno por aprender más y así superar a sus pares fue provechosa. QueryCompetition sirvió como un punto de convergencia para el aprendizaje individual y grupal a través de una sana competencia.

Como establece [6], la utilización de métodos de evaluación que sean innovadores proporciona un punto extra de motivación al alumnado, lo que facilita la docencia. Por otra parte, siempre es grato el reconocimiento a los logros de los alumnos, en este caso el hecho de llegar a una etapa final de un concurso y ganarlo, lo cual resultó gratificante para muchos de los participantes.

Como trabajo futuro, QueryCompetition será utilizado en más grupos y se socializará para que profesores o alumnos interesados hagan uso de esta herramienta didáctica. Paralelamente, se construirá una nueva versión atendiendo las oportunidades de mejora identificadas.

## Referencias

- [1] O. Colomina, P. Compañ, R. Satorre, F. Aznar, P. Suau and R. Rizo. Aprendiendo mediante juegos: Experiencia de una competición de juegos inteligentes. In *Actas de las X Jornadas de Enseñanza Universitaria de Informática*. pp. 513-515, Alicante, España. 2004.
- [2] J.S. Eccles, T.F. Adler, R. Futterman, S.B. Goff, C.M. Kaczala, J.L. Meece and C. Midgley. Ex-

- pectancias, values and academic behaviors. In J.T. Spence (Ed.), *Achievement and achievement motivation*. San Francisco, CA, EUA. 1983.
- [3] L. García-Mundo, J. Vargas-Enriquez, M. Genaro and M. Piattini. ¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática? In *Actas de las XX Jornadas de Enseñanza Universitaria de Informática*. pp. 303-310, Oviedo, España. 2014.
- [4] M. Gardner. ¡Ajá! Paradojas. Paradojas que Hacen Pensar. Ed. Labor, S.A. 1989.
- [5] J.A. Gómez, J. Planells, A. Casanova, M. Galiano, M. Llorens, G. Moltó, F. Marqués, N. Prieto, F. Álvaro, A. Barella, M.J. Castro-Bleda, M. García, A. Hermida, C. Herrero, C.D. Martínez-Hinarejos, A. Molina, L. Moreno, J. Pastor, M. Pastor, M. Rodríguez, A. Terrasa and E. Vivancos. Competiciones de programación. Estímulo y salida laboral para los alumnos. In *Actas de las XIX Jornadas de Enseñanza Universitaria de Informática*. pp. 161-166, Castellón, España. 2013.
- [6] A. Gómez-Rodríguez, J.C. González-Moreno, D. Ramos-Valcárcel and F.J. Rodríguez-Martínez. Evaluación de competencias en Ingeniería de Software mediante competición. In *Actas de las XVIII Jornadas de Enseñanza Universitaria de Informática*. pp. 137-144, Ciudad Real, España. 2012.
- [7] J. Hopcroft, R. Motwani and J. Ullman. Introducción a la teoría de autómatas, lenguajes y computación. Pearson Addison-Wesley. ISBN: 978-84-7829-088-8. 1979.
- [8] G. Jimenez-Díaz, J. Recio-García, B. Díaz-Agudo and G. Flores-Puga. Uso de competiciones y sistemas de clasificación como metodología de evaluación de una asignatura. In *Actas de las XVIII Jornadas de Enseñanza Universitaria de Informática*. pp. 25-32, Ciudad Real, España. 2012.
- [9] J.C. Núñez. Motivación, Aprendizaje y Rendimiento Académico. In *Actas del X Congreso Internacional Galego-Português de Psicopedagogia*. Braga, Portugal. ISBN: 978-972-8746-71-1. 2009.
- [10] H.G. Rice. Classes of Recursively Enumerable Sets and Their Decision Problems. *Trans. Amer. Math. Soc.* 74, 358-366, 1953.