

ENTORNO WEB PARA LA GENERACIÓN DINÁMICA DE REACTIVOS EN LA EVALUACIÓN DE APLICACIÓN DE DERIVADAS

Rey David Lagunas Antonio¹, Luis Ángel Reyes Hernández²

¹Licenciado en Ciencias de la Computación, ²Maestro en Sistemas Computacionales

^{1,2}División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Orizaba, Veracruz,
México

¹rey_david2@hotmail.com, ²l.a.reyes.h@gmail.com

Abstract

El presente artículo propone una aplicación Web dirigida al sector educativo utilizando como caso de estudio la materia de Cálculo Diferencial, del cual se toma como principal punto el tema titulado Aplicación de Derivadas impartida en todas las carreras del Instituto Tecnológico de Orizaba, donde existe un alto índice de reprobación ya que los tiempos dedicados a ese tema no es suficiente. El objetivo es proporcionar una aplicación tipo Quiz empleando un algoritmo de tipo MCQ (Multiple Choice Question) para generar de forma aleatoria cuestionarios en donde las opciones de respuestas correctas siempre serán diferentes, ya que los reactivos tendrán variables y parámetros de entrada generados aleatoriamente dentro de rangos establecidos desde un principio. La aplicación Web, desarrollada en JSF 2.0 (Java Server Faces), evaluará cada reactivo mediante el algoritmo para cada pregunta y se almacenará todo en un repositorio de información. Además, se busca que la aplicación Web sea fácilmente mantenible por otros desarrolladores aplicando el patrón arquitectónico MVC (Modelo Vista Controlador).

I. INTRODUCCIÓN

En todas las Ingenierías ofertadas en el Instituto Tecnológico de Orizaba se imparte la materia de Cálculo Diferencial, en dicha asignatura existe un tema titulado "Aplicación de derivadas", el cual hay ocasiones que el tiempo estimado en el seguimiento del curso queda inconcluso y a veces sin llegar a ser visto evaluándolo con un trabajo. El problema que se identifica es la falta de tiempo asignada al temario, inclusive falta de tiempo a la aplicación de las derivadas, aún cuando son parte medular de toda ingeniería ofertada en el Instituto Tecnológico de Orizaba, como consecuencia el alumno carece de visión para aplicar dichas derivadas en situaciones o ejemplos propios de su especialidad. Dicha materia es fundamental para cualquier Ingeniería ofertada en el Instituto Tecnológico de Orizaba y es de vital importancia que los conocimientos adquiridos sean comprendidos para posteriores materias que lo requieran.

El proceso de enseñanza-aprendizaje exige cada vez más ofrecer al alumno maneras diversas para adquisición de conocimiento. El *e-learning* se presenta como una de las estrategias formativas que resuelve muchos de los problemas educativos. Se propone desarrollar una aplicación Web que utilice un repositorio de preguntas adecuados al tema titulado "Aplicación de derivadas" para ser utilizado por los alumnos del Instituto Tecnológico de Orizaba, con el

fin de reforzar sus conocimientos de Cálculo Diferencial y su habilidad para interpretar y resolver problemas específicos de esta disciplina.

La introducción de elementos aleatorios en la generación de cuestionarios es una característica valorada por los autores. Esto se hace más patente, si cabe, en el caso de disciplinas científico-técnicas como las matemáticas, física o química. En estas materias es posible generar cuestiones con un enunciado aleatorio que permiten al estudiante ejercitar sus habilidades. La idea principal es que el Problemario sea dinámico, es decir, los reactivos contienen variables y parámetros de entrada que se generan de manera aleatoria (dentro de ciertos rangos previamente establecidos), de modo que las opciones de respuestas correctas y los distractores (opciones de respuestas incorrectas) siempre serán diferentes. La computadora evaluará cada reactivo a través del algoritmo tipo MCQ [1] y se almacenará todo en una base de datos.

II. TRABAJOS RELACIONADOS

El número de estudiantes que ingresan a la Universidad Nigeriana aumenta rápidamente, los salones están sobresaturados de alumnos. Según se discute en [2], atender con calidad a dichos estudiantes implica la construcción de más universidades y contratación de más profesores. Se propuso como solución el diseño y desarrollo de una aplicación *Web*, utilizando tecnología *.NET*, que ayude al proceso de evaluación de los alumnos, mediante exámenes de pregunta de selección múltiple a través de *Internet*, calificados automáticamente e inmediatamente después de responder la evaluación. Se concluyó que la aplicación *Web* fue una solución al problema de llevar a cabo evaluaciones para grupos numerosos de estudiantes. El profesor se despreocupa de estar calificando una multitud de exámenes. Además, la Universidad Nigeriana logró actualizarse tecnológicamente con respecto a la educación.

Las universidades de Reino Unido frecuentemente utilizan los exámenes asistidos por computadora porque evalúa rápidamente y eficientemente a un gran número de estudiantes al mismo tiempo, grupos de hasta 300 alumnos o más. Sin embargo, algunas universidades solamente disponen de salones para albergar a 100 alumnos. Una posible solución sería dividir al grupo de estudiantes en subgrupos con menor número de estudiantes y evitar la comunicación entre estudiantes; evitar la comunicación entre estudiantes es poco práctico. Una alternativa de solución es utilizar un *software* para generar versiones aleatorias de un examen, a partir de un banco de preguntas organizadas por tema y nivel. De este modo, paralelamente, cada estudiante recibe una versión equivalente del examen. Sin embargo, profesores y estudiantes no están de acuerdo en la selección aleatoria de preguntas para formar versiones aleatorias de exámenes, pues afirman que es difícil asegurar que los exámenes sean totalmente equivalentes. En [3] se propone utilizar un banco de preguntas calibradas y un método para calcular las calificaciones con base al grado de dificultad de las preguntas que el estudiante recibió; este método se llama OSIRIS (*Objective Standardisation in Random Item Selection*, Estandarización Objetiva de la Selección de Datos al Azar). Se concluyó que OSIRIS, junto con la utilización de las mejores prácticas para el aseguramiento de la calidad en el banco de preguntas, podría mejorar la confianza en los exámenes generados aleatoriamente, en vez, de ver la aleatoriedad como un riesgo.

Desarrollar exámenes en línea (*e-assessments*) tiene un costo en tiempo y esfuerzo. Existen investigaciones enfocadas a la generación automática o semi-automática de preguntas a través de un modelo (plantilla) de pregunta. Se propone un proceso para la generación de preguntas de tipo MCQ simple a partir de la información proporcionada, mediante ontologías, por DBpedia (una fuente de datos). En [4] se siguió el estándar IMS-QTI (*IMS Question and Test Interoperability Specification*, Especificación de Interoperabilidad de Preguntas y Pruebas de IMS) para que las preguntas sean importadas en la mayoría de las plataformas de *e-learning*, permitiendo que las preguntas sean portables. Se evaluaron cinco modelos con el fin de identificar las principales problemáticas que tienen las fuentes de datos para apoyar la generación de preguntas. Se concluyó que las preguntas generadas no eran de calidad, ya que el 23.33% de las preguntas tuvieron defectos en respuestas correctas; el 33.33%, tuvo defectos en los distractores; solamente el 53.33% de las preguntas no tuvo ningún tipo de defecto. El rendimiento del sistema, comparado con otros enfoques, mostró su potencial como una estrategia para generar preguntas de exámenes. Fue compleja la generación automática de las preguntas. La *Web* semántica proporciona modelos de conocimiento a partir del cual se extrae la información para generar varias instancias de un modelo de pregunta.

Existen investigaciones que se enfocan a evaluar la comprensión de textos mediante preguntas MCQ, dando mayor relevancia a la respuesta correcta y generando arbitrariamente los distractores. Se describe y evalúa DQGen (*Diagnostic Question Generator*, Generador de Preguntas de Diagnóstico) un sistema que utiliza procesamiento de lenguaje natural para generar preguntas que diagnostique los distintos tipos de deficiencia en la comprensión de textos para niños. En [5] se proporciona una opción correcta y tres distractores por cada pregunta. Pero cada uno de estos distractores se clasifica en error sintáctico, semántico, por mencionar algunos, para determinar el tipo de deficiencia lectora del niño. Las preguntas generadas por DQGen fueron evaluadas por miembros del equipo de investigación. Se concluyó que en promedio las preguntas resultaron de buena calidad; sin embargo, algunas preguntas se podían responder sin necesidad de leer el texto. No hubo problema para clasificar las respuestas como correctas, pero hubo inconvenientes al clasificar los distractores en semánticos, se requieren más restricciones para hacer una clasificación adecuada de los distractores.

Existen diferentes modelos computacionales para generar analogías de la forma "A es a B como C es a D". Sin embargo, la generación de analogías presenta inconvenientes que requieren más investigación. Se presenta un nuevo enfoque para generar analogías en formato MCQ, estas analogías se utilizan para evaluar a los estudiantes. Se propone utilizar ontologías existentes de alta calidad para obtener los datos para crear las analogías de manera automática. En particular, en [6] se utilizaron tres ontologías: *Gene Ontology*, *People and Pets* y *Pizza Ontology*. Se describe las características de una buena pregunta de analogía. Se propuso un agente que resolviera las preguntas para probar el algoritmo de creación de preguntas de analogías de selección múltiple. Este agente resolvió 8% de las preguntas generadas a partir de *Gene Ontology*; 67%, de *People and Pets*; 88%, de *Pizza Ontology*. La terminología utilizada en *Gene Ontology* repercutió en el porcentaje de solución por parte del agente, pues utiliza una terminología muy específica. Se argumenta que la dificultad del dominio es un factor que dificulta controlar la generación y solución automática de las preguntas.

En [7] se establece que Moodle, plataforma *e-learning*, se permite crear cuestionarios. Sin embargo, es tedioso ingresar un número grande de preguntas, existe la posibilidad de cometer errores de digitación, no posee un editor de LaTeX, solamente se generan preguntas aleatorias para las preguntas de tipo Calculada (útil para profesores de matemáticas). *ScientificWorkPlace* genera cuestionarios específicos a partir de plantillas de preguntas. Sin embargo, Moodle no da soporte al tipo de formato de dicha plantilla. Se propone un algoritmo para convertir exámenes realizados en *ScientificWorkPlace* en formato GIFT (*General Import Format Technology*, Tecnología de Formato General de Importación) para Moodle; se presenta un prototipo (Tex2Gift) para validar el algoritmo. Además, se propone un protocolo para generar preguntas aleatorias para Moodle, utilizando *ScientificWorkPlace*. Tex2Gift generó aleatoriamente preguntas MCQ simple en formato GIFT a partir de una plantilla tex de *ScientificWorkPlace*, esta conversión fue automática; por tanto, el profesor, solamente se enfocó en escribir preguntas genéricas en plantillas *ScientificWorkPlace*.

En la tabla 1 se da una perspectiva general del tema que tratan los trabajos relacionados. De los trabajos relacionados, se distingue que existen múltiples propuestas de desarrollo de software asociados con el proceso de evaluación. Existen diferentes enfoques para dar solución a las diferentes problemáticas, desde la utilización de lenguaje natural hasta la utilización de ontologías para la generación de exámenes aleatorios; otros autores, se enfocan a que el *software* genere exámenes que sean interoperables entre diferentes plataformas de *software* educativo utilizando diferentes tecnologías, como *.Net*, por mencionar algunas.

Tabla 1. Trabajos relacionados 1 1

Artículo	Tecnología	Estado	Nivel
Development of an E-Assessment Platform for Nigerian Universities. O. Akinsanmi, et al [2]	.NET	Terminado	Superior
In search of Osiris: random item selection, fairness and defensibility in high-stakes e-assessment. J. Dermo [3]	No aplica	Terminado	No aplica
Generating educational assessment items from linked open data: the case of DBpedia. Muriel Foulonneau [4]	DBpedia	Prototipo	No aplica
Generating diagnostic multiple choice comprehension cloze questions. J. Mostow y H. Jang [5]	No especifica	Terminado	Para niños
Automatic generation of analogy questions for student assessment: an Ontology-based approach. T. Alsubait, et al [6]	No especifica	Terminado	General
Generación de Preguntas Aleatorias de Opción Múltiple con Única Respuesta para Moodle. F.A. Villa, J.A. Jiménez, y R. Franco [7]	Scientific Workplace, C# .NET	Prototipo	Superior

III. PROPUESTA

Una de las principales necesidades a considerar al desarrollar la presente aplicación *Web* es que sea capaz de evolucionar para posteriores versiones del *software*, es decir, que sea fácil de dar mantenimiento, que permita modificar tan solo porciones (módulos) de la aplicación sin necesidad de modificar toda la aplicación. El patrón arquitectónico Modelo Vista Controlador (MVC) permite desacoplar el modelo (datos) de la vista (presentación de datos), dar fácil mantenimiento, permite crear aplicaciones escalables y permite verificar y validar módulos de la aplicación. Cabe mencionar que la variación de MVC para aplicaciones se llama MVC Modelo 2 [8]. Con base a lo mencionado, se propone como una alternativa de solución aplicar MVC, la cual se muestra en la fig. 1. En dicha figura, se muestra que la aplicación *Web* se separa en tres capas; en la capa vista, se representa a la interfaz de usuario con archivos XHTML, además se propone utilizar la biblioteca *PrimeFaces* para permitir la utilización de *Ajax* con *JSF* [9,10]. A través del *servlet Faces* se gestionan las peticiones de recursos a la aplicación *Web* y las respuestas a mostrar en la vista, a través del respectivo *Managed Bean* (son responsables del Modelo), el cual almacena los datos a mostrar en las vistas o los recuperadas de la entrada de usuario; el *Managed Bean* administra la lógica de la aplicación delegándolo al *Model Bean* (clases del dominio). El *Model Bean*, entre otros, manipula la información contenida en el sistema gestor de base de datos utilizando la biblioteca de conexión *JDBC*.

IV. CONCLUSIONES

El uso de las aplicaciones *Web* se ha incrementado y con ello la necesidad de contar cada día con mejores aplicaciones; en el caso especial del Instituto Tecnológico de Orizaba en la materia que imparte de Cálculo Diferencial hay un tema en específico denominado Aplicación de Derivadas que es la parte medular para toda ingeniería. La evaluación de este tema es un proceso que requiere de considerable tiempo por parte del docente. Se propone el desarrollo de una aplicación *Web* que permite generar versiones de exámenes aleatorios con el fin de hacer más eficiente el proceso de evaluación, ya que la aplicación presentará el resultado correcto ahorrando trabajo y tiempo al docente. De los trabajos relacionados descritos, se observa que existen software que genera exámenes de manera automatizada utilizando diferentes enfoques o algoritmos, hay autores que utilizan ontologías o lenguaje de procesamiento natural; otros, garantizan diferentes versiones de exámenes equivalentes calibrando las preguntas. Sin embargo, ningún trabajo se enfoca en los distractores de las respuestas posibles. Por otro lado, para solventar las necesidades de que la aplicación *Web* sea fácilmente mantenible, es decir, que se facilite la incorporación de posteriores cambios al *software* según se requiera, se propuso como una alternativa utilizar la arquitectura MVC, la cual permite el desacoplamiento de la capa de presentación de la de modelo.

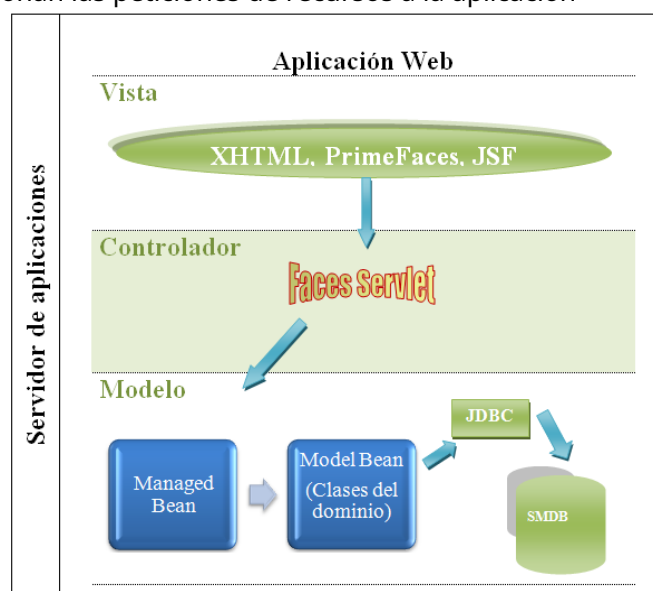


Figura 1. Arquitectura MVC propuesta 1

V. TRABAJO A FUTURO

Diseñar una aplicación *Web* para generar exámenes de manera aleatoria tipo *Quiz*[11], conformado por preguntas de selección múltiple simple; inclusive el profesor podrá añadir algún ejercicio si así lo ve pertinente para evaluar al alumno. En concreto, se busca conocer y aplicar repositorios de preguntas de Cálculo Diferencial; aplicar un algoritmo MCQ para la construcción de la aplicación *Web*; identificar y aplicar una metodología ágil para el desarrollo y sustento de la aplicación; aplicar el lenguaje de programación, JSF, para el desarrollo de la aplicación; dar a conocer la aplicación de la herramienta; aplicar un caso de estudio para analizar los resultados

REFERENCIAS

1. Sandra Castañeda Figueiras, *Educación, aprendizaje y cognición. Teoría en la Práctica. Guadalajara*, México: El Manual Moderno, 2004.
2. O. Akinsanmi, et al. "Development of an E-Assessment Platform for Nigerian Universities," *Research Journal Applied Sciences, Engineering and Technology*, vol. 2, no. 2, pp. 170-175, 2010
3. J. Dermo, "In search of Osiris: random item selection, fairness and defensibility in high-stakes e-assessment," *International Journal of e-Assessment*, vol. 1, no. 1, 2011.
4. Muriel Foulonneau. "Generating educational assessment items from linked open data: the case of DBpedia," In *Proc. of the 8th international conference on The Semantic Web(ESWC'11)*, RaúlGarcía-Castro, Dieter Fensel, and Grigoris Antoniou (Eds.). Springer-Verlag, Berlin, Heidelberg, 2011, pp. 16-27. DOI=10.1007/978-3-642-25953-1_2 http://dx.doi.org/10.1007/978-3-642-25953-1_2
5. Jack Mostow y Hyeju Jang. "Generating diagnostic multiple choice comprehension cloze questions". In *Proc. of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp.136-146.
6. T. Alsubait, et al. , "Automatic generation of analogy questions for student assessment: an Ontology-based approach," *Research in Learning Technology*, vol. 20, pp. 95-101, Agosto 2012
7. F.A. Villa, J.A. Jiménez, y R. Franco. "Generación de Preguntas Aleatorias de Opción Múltiple con Única Respuesta para Moodle". *Lámpsakos*, vol. 3, no. 6, pp 32-37. Jul-Dic 2011
8. Giulio Zambon, *Beginning JSP, JSF and Tomcat Java Web Development*, 2nd ed. New York, USA: Apress, 2012.
9. D. Geary and C. Horstmann, *Core JavaServer Faces*. CA, USA: Pearson Education, 2010, p. 3,13.
10. E. Burns and C. Schalk, *The complete reference JavaServer Faces 2.0*. NY, USA: McGraw-Hill, 2010, pp.3,6,7.
11. A. Llamas Ruiz, "*QUIZ LIBRE: Software de entretenimiento basado en un juego de preguntas y respuestas para plataforma Nintendo DS*". Memoria de Carrera, Escuela Superior de Ingeniería Universidad de Cádiz, España, 2012.