

Sistema gestor de exámenes: una alternativa digital a los exámenes tradicionales

Manuel Alejandro Valdés-Marrero*, Remedios Fabián-Velasco
Saúl Gómez-Carretero, Christian Ayala-Esquivel
& Ángel Salvador López-Vásquez

Resumen

El objetivo de este trabajo es presentar un sistema de *software* funcional exclusivamente de gestión de exámenes, confiable, bilingüe, gratuito e intuitivo, que busca imitar en la medida de lo posible a la aplicación tradicional de exámenes. En la educación, los exámenes son la herramienta más común para evaluar el aprendizaje de los estudiantes. El sistema presentado en este artículo pretende ser una opción de herramienta para aquellos profesores que buscan una alternativa digital que cumpla con los requerimientos de los exámenes tradicionales. La disponibilidad de internet o de una red local no es requisito indispensable para utilizar el sistema, por lo que los exámenes pueden ser aplicados en cualquier instalación que cuente con equipos de cómputo o con los equipos de cómputo portátiles de los estudiantes. También cuenta con medidas de protección para que los estudiantes no pierdan sus avances por problemas con el suministro eléctrico. El sistema permite la creación y utilización de bancos de preguntas por materias, para fomentar el trabajo colegiado de profesores por academias o departamentos, lo que permitiría generar exámenes departamentales o estandarizados. Adicional al objetivo principal, se presenta cómo se desarrolló el sistema, lo cual puede ser de utilidad para desarrolladores o estudiantes de ingeniería de software que requieran un ejemplo del desarrollo de un sistema funcional. El sistema ha sido probado con estudiantes no habituados al uso de software, obteniendo resultados satisfactorios. El sistema está actualmente operativo y se han aplicado más de 200 exámenes con él.

Palabras clave: desarrollo de *software*, examen digital, gestión de exámenes, preguntas cerradas.

Recibido: 26 de septiembre de 2022.

Abstract

The objective of this work is to present a functional, reliable, bilingual, free, and intuitive *software* system solely for test management, which seeks to imitate, to the extent possible, the traditional test taking. In education, tests are the most common tool to assess the students' learning. The system presented in this article is meant to be an option of a tool for those teachers who seek a digital alternative which fulfil the requirements of traditional testing. Internet or local network availability is not an essential requirement for using the system, so testing may be performed in any installation that has computers or with the students' portable computers. It also has protective countermeasures so that students don't lose their progress due to energy supply problems. The system allows the creation and usage of question banks by course of study, to encourage teachers' collegial work by academies or departments, which can be used to generate standard or departmental tests. In addition of the main objective, how the system was developed is presented, which can be useful for developers or software engineering students who require an example of a functional system development. The system has been tested with students not accustomed to software usage, obtaining satisfactory results. The system is currently operational and over 200 tests have been applied with it.

Key words: *software* development, digital test, test management, closed questions.

Aceptado: 06 de diciembre de 2022.

¹ Instituto de Industrias, Universidad del Mar campus Puerto Escondido. Ciudad Universitaria, Vía Sola de Vega km 2.5 Carretera Puerto Escondido- Oaxaca. San Pedro Mixtepec-Juquila, Oaxaca, México, 71980.

* Autor de correspondencia: valdes@zicatela.umar.mx

Introducción

Los exámenes son instrumentos de evaluación que se emplean para identificar el nivel de dominio de los sustentantes sobre un constructo específico (INEE 2019). Ejemplos de exámenes estandarizados en educación básica y media superior son las pruebas ENLACE (Evaluación Nacional del Logro Académico en Centros Escolares) y EXCALE (Exámenes de la Calidad y el Logro Educativo), administradas por la SEP (Secretaría de Educación Pública) y el INEE (Instituto Nacional para la Evaluación de la Educación), respectivamente. Estos exámenes y evaluaciones fueron analizados y mejorados, dando paso a PLANEA (Plan Nacional para la Evaluación de los Aprendizajes), el cual tiene dos modalidades de evaluación: PLANEA SEN, para dar información al nivel del Sistema Educativo Nacional en los distintos niveles de la educación obligatoria, y PLANEA Escuelas, para ofrecer información al nivel de centros escolares. En ambas modalidades, se contempla a los exámenes como herramienta de evaluación diagnóstica para alcanzar sus objetivos (INEE 2018). En la educación, se recomienda que los exámenes se complementen con otras actividades para generar una evaluación en el aula, por ejemplo: exposiciones, resolución de casos, investigaciones, actividades realizadas en clase y proyectos (Basogain-Urrutia 2021). En casos como las evaluaciones certificativas, donde no importa el proceso de enseñanza-aprendizaje sino solo los resultados, el examen es el mejor instrumento de evaluación (Otero-Doval 2014).

En el caso de estudio específico de la Universidad del Mar, en su Reglamento vigente de Alumnos –en el Título V, Capítulo I “de la evaluación”– precisa a la evaluación como el comprobante del nivel de aprendizaje y desempeño académico

del estudiante, otorgando el 50% al mecanismo de examen escrito del total de las calificaciones parciales y ordinarias, y el 100% en las calificaciones extraordinarias y especiales. Los exámenes escritos ordinarios, extraordinarios y especiales se entregan a resguardo en el Departamento de Servicios Escolares, en periodos establecidos (SUNEO 2016). Este último requerimiento normativo hace necesario que se elaboren los exámenes en papel, lo cual tiene una serie de inconvenientes desde diferentes puntos de vista:

- Económico. El uso de exámenes en papel conlleva un gasto considerable para la Universidad, el profesorado e incluso los estudiantes, ya sea en ediciones, impresiones, copias, hojas blancas, almacenaje, conservación, etc.
- Ambiental. El uso excesivo del papel representa hoy en día un gran problema para el medio ambiente, principalmente por los recursos naturales que se necesitan para producir dicho papel y, en menor medida, por la contaminación que genera cuando se desecha; de acuerdo con el Centro de Ciudadanía Corporativa (CCC) de la Fundación de la Cámara de Comercio de los Estados Unidos, en 2016, el papel se consideraba la categoría de desperdicio más grande de las empresas (Bowdish 2016).
- Académico. Los profesores deben calificar los exámenes tradicionales de forma manual, lo cual lleva tiempo y, dependiendo el número de estudiantes, representa una carga adicional de trabajo para aquellos profesores que imparten clases a grupos numerosos; la consecuencia es que los estudiantes deben esperar para recibir el resultado de su evaluación.
- Pedagógico. La calidad de cualquier examen ya sea en papel o basado en computadora, se mide en función de la

validez, la confiabilidad y la imparcialidad; en otras palabras, un examen elaborado correctamente debe evaluar los conocimientos, habilidades y capacidades apropiados, y debe hacerlo de manera consistente para cada persona evaluada y debe evitar cualquier sesgo que pueda empañar los resultados (Prometric 2021). Sin embargo, siempre existe la posibilidad de que haya sesgo en la calificación de exámenes en papel, puesto que el profesor evalúa directamente las respuestas y puede verse influenciado por la identidad del estudiante, por errores previos, por cambio de criterios o humor, por cansancio, o no seguir una técnica adecuada para evaluar.

- Ético. La aplicación de los exámenes tradicionales presenta el problema referente a la posibilidad que tienen los estudiantes para copiar, aun cuando estén bajo la supervisión de un profesor. Esta posibilidad se materializa con el uso de “acordeones”, descifrar patrones de generación de respuestas en los exámenes, falta de honestidad y el uso de claves entre compañeros, los cuales son ejemplos claros de la vulnerabilidad que poseen los exámenes tradicionales ante esta situación.

- Administrativo. Manejar los exámenes en papel no es una tarea sencilla, ya que existen riesgos importantes, como la posibilidad de perder el examen de algún estudiante y no tener un medio para recuperarlo, o en el peor de los casos, extrañar un examen y que este llegue a manos equivocadas. Igualmente, si no se realiza y utiliza un banco de preguntas, que constantemente se enriquezca con nuevas preguntas, cada semestre se tiene que rehacer el trabajo desde el inicio o se corre el riesgo de preparar a los estudiantes solo para los temas de un examen predefinido. Los profesores que se integran a la planta docente no cuentan con algún punto de partida

para elaborar sus propios exámenes.

Ante todas estas situaciones, los profesores han intentado migrar a la tecnología web, utilizando *software* gratuito para elaborar, aplicar y calificar exámenes. Hay pocos sistemas enfocados solo a la elaboración y revisión de exámenes, más bien, son sistemas completos de gestión de aprendizaje (LMS), sistemas de gestión de contenidos (CMS) o una combinación de ellos (LCMS) (Watson & Watson 2007). Ejemplos en el mercado de estos sistemas enfocados a educación son Moodle, Blackboard Learn o Google Classroom (Basogain-Urrutia 2021). Sin embargo, estos sistemas pueden incluir funcionalidades que no sean del interés del profesor y causen confusión al utilizarlos. Existen algunas dificultades al utilizar este tipo de sistemas:

- Tecnología. Por ser sistemas basados en red e Internet, se necesitan espacios adicionales (salas de cómputo) para utilizar estos sistemas, ya que muchos espacios físicos están desconectados o restringidos (en el caso de estudio, la sala de autoacceso de la UMAR campus Puerto Escondido no cuenta con el servicio de internet, ni red dentro de sus instalaciones; lo mismo con la mayoría de las aulas).

- Calificación automática. No es muy confiable, especialmente en preguntas de rellenar espacios, que requieren un análisis léxico concreto. Cuando la respuesta del estudiante difiere con respecto a lo establecido por el profesor en alguna letra, si hay una diferencia de uso entre letras mayúsculas o minúsculas, cuando se agregan espacios adicionales en la respuesta, o si se utiliza algún sinónimo, los sistemas la consideran una respuesta incorrecta. Por lo que el profesor tendrá la necesidad de plantear múltiples respuestas correctas o calificar manualmente las preguntas y corregir la evaluación del sistema.

- **Recuperación.** No todos los sistemas cuentan con mecanismos de recuperación para conservar el avance que los estudiantes llevan al contestar un examen. Esto implica que, si hay problemas de red o de suministro eléctrico, el estudiante deberá empezar de nuevo, probablemente reiniciando el sistema o incluso el equipo, con la pérdida de tiempo que esto podría ocasionar.
- **Navegación y corrección.** No todos los sistemas permiten que el estudiante navegue libremente por el examen, que conteste las preguntas en el orden que considere adecuado, que le permita saltar preguntas o incluso corregirlas, como se haría en la mayoría de los exámenes tradicionales.
- **Idioma.** El hecho de no tener una herramienta bilingüe limita las posibilidades para realizar los exámenes, ya sea porque el *software* con el que se quiere trabajar se encuentra en inglés y se necesita uno en español, porque el *software* usa traducción automática y no se entienda su contenido, o por la necesidad de realizar exámenes en inglés y el *software* solo soporta el idioma español.

Por todo lo anterior, se desarrolló un sistema bilingüe (español - inglés) dedicado específicamente a la elaboración y revisión de exámenes digitales (evitando así el uso de papel); con la capacidad de trabajar con varios tipos de preguntas cerradas que serán calificadas automáticamente (para de esta manera eliminar el sesgo); con la posibilidad de crear exámenes aleatorios a partir de conjuntos de respuestas y preguntas aleatorias almacenadas en un banco de preguntas para cada una de las materias (de esta manera se disminuye el riesgo de que los estudiantes apliquen algún tipo de trampa en el examen); que cuenta con mecanismos de seguridad y de recuperación, como lo son el guardado automático de avance de exámenes, así

como el respaldo en PDF de exámenes en blanco, respuestas de los exámenes y los exámenes contestados por los estudiantes. El sistema no requiere de ningún tipo de conexión a internet o alguna red interna. La aplicación para presentar los exámenes simula el comportamiento de los exámenes tradicionales al permitir a los estudiantes navegar entre preguntas, marcarlas y corregirlas antes de entregar el examen; y se apega al reglamento de alumnos vigente de la UMAR en lo referente a evaluaciones, ya que el contenido de los exámenes está en archivos digitales que pueden reconstruir el examen físico o digital en caso de alguna revisión o la solicitud de copia por parte del estudiante.

A este sistema se le denominó SIGEXAM, "Sistema gestor de exámenes UMAR", en el resto del artículo se pretende describir su uso y desarrollo.

SIGEXAM como producto de *software*

SIGEXAM, un sistema de información no basado en red o internet para elaborar, aplicar y calificar exámenes, enfocado a la Universidad del Mar. SIGEXAM cuenta con dos subsistemas, uno para el profesor y otro para los alumnos. Como debe ser en cualquier producto de *software* (Sommerville 2011), cada subsistema cuenta con un disco de instalación y documentación de usuario, en este caso en forma de un manual de instalación y un manual de introducción. A continuación, se resume como interactúan los subsistemas de SIGEXAM.

SIGEXAM subsistema profesor (Fig. 1) permite al profesor diseñar exámenes con diferentes tipos de preguntas cerradas: opción múltiple (OM), relacionar columnas (RC) y rellenar espacios (RE).

- Las preguntas tipo OM permiten



Figura 1. SIGEXAM Subsistema profesor. Menú principal.

establecer una pregunta y una serie de respuestas, las cuales son marcadas por el profesor como correctas o incorrectas. Una pregunta puede tener una o más respuestas correctas. Cuando el estudiante visualice la pregunta, el sistema le indicará cuántas respuestas debe marcar, si hubiera más de una (Fig. 2). El sistema da crédito parcial por cada respuesta correcta. Con este tipo de preguntas se pueden hacer reactivos de falso/verdadero, así como de cuestionamiento directo, elección de elementos y de ordenamiento (Ceneval 2021).

- Las preguntas tipo RC permiten establecer un texto de introducción, y una serie de parejas pregunta/respuesta. Puede haber preguntas sin respuesta o inclusive una sola respuesta sin pregunta. Cuando el estudiante visualice la pregunta, el sistema le mostrará por incisos las preguntas y deberá seleccionar dicho inciso en la respuesta mediante listas desplegables; también le indicará si hay preguntas sin respuesta, si es el caso (Fig. 3). El sistema da crédito parcial por cada pareja pregunta/respuesta correcta.

- Las preguntas tipo RE permiten establecer una pregunta con marcadores para indicar texto faltante y una serie de textos para poner en los espacios. Mediante números, el profesor indica los textos

correspondientes a cada espacio, siendo válido que haya textos que no van en ningún espacio. Todos los espacios deben tener una y solo una respuesta. Cuando el estudiante visualice la pregunta, el sistema le mostrará el texto original y una serie de espacios para colocar el texto faltante. Los textos faltantes son mostrados abajo, donde pueden ser arrastrados y soltados por el estudiante en los espacios correspondientes (Fig. 4). El sistema da crédito parcial por cada respuesta correcta.

La gestión de los exámenes requiere datos del profesor, campus y carreras, así como de las materias, cursos y las preguntas que se realicen se deben clasificar en temas y subtemas. Junto con los cursos se agrega la lista de alumnos (Fig. 5), la cual se puede escribir o importar.



Figura 2. Pregunta de opción múltiple (OM).



Figura 3. Pregunta de relacionar columnas (RC).

Posteriormente, en el catálogo de exámenes (Fig. 6) se permite generar dos tipos de exámenes: manual y aleatorio. En un examen de tipo manual, se debe agregar pregunta por pregunta o todas las preguntas de un tema y subtema, después pueden ordenarse de forma manual. En un examen de tipo aleatorio, se debe indicar el número de exámenes diferentes y, en vez de seleccionar preguntas específicas, se selecciona cada tema y subtema a incluir en el examen, más la cantidad y tipo de pregunta a agregar (OM, RC, RE). En la medida de lo posible, el sistema tratará de tomar preguntas diferentes, dependiendo de la cantidad de preguntas de cada tipo que haya almacenadas para cada subtema en particular dentro del banco de preguntas. El orden final de las preguntas en el examen será aleatorio. En ambos tipos de exámenes, las respuestas a todas las preguntas serán presentadas en orden aleatorio cuando el estudiante visualice el examen.

El sistema pedirá la ponderación de cada pregunta al guardar el examen, es decir, cuántos puntos vale cada pregunta. Los exámenes se guardan en un formato propio del sistema dentro de un archivo binario con extensión EXAM (Fig. 7). Este es el examen portable para compartir con los estudiantes. En la modalidad de aleatorio, se generará la cantidad de exámenes diferentes que haya sido solicitada. Una vez diseñados los exámenes, se pueden eliminar completamente o modificarlos, agregando o quitando preguntas u ordenándolas. Al guardar los cambios se vuelve a pedir la ponderación de las preguntas. También se puede imprimir los exámenes en *PDF*, en blanco o contestados.

SIGEXAM subsistema alumno sirve para contestar los exámenes generados en el subsistema profesor. Para iniciar un examen, el estudiante escribe su nombre y



Figura 4. Pregunta de rellenar espacios (RE).

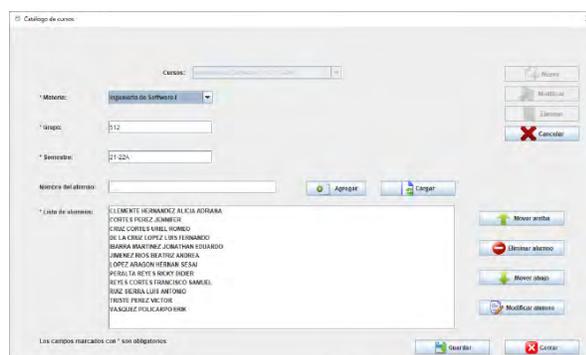


Figura 5. Subsistema profesor. Agregar nuevos alumnos a un curso.

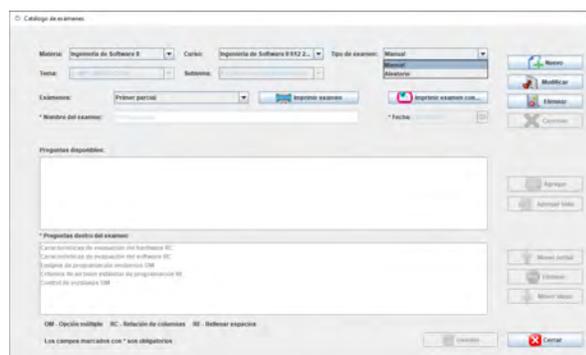


Figura 6. Subsistema profesor. Ventana del catálogo de exámenes.

selecciona el archivo con extensión EXAM (Fig. 8). Dado que la extensión es ligada con la aplicación durante la instalación de SIGEXAM alumno, también se puede abrir la aplicación y seleccionar automáticamente el examen, dando doble clic sobre el archivo del examen desde el explorador de archivos, como cualquier otro archivo reconocido por el sistema operativo Windows.

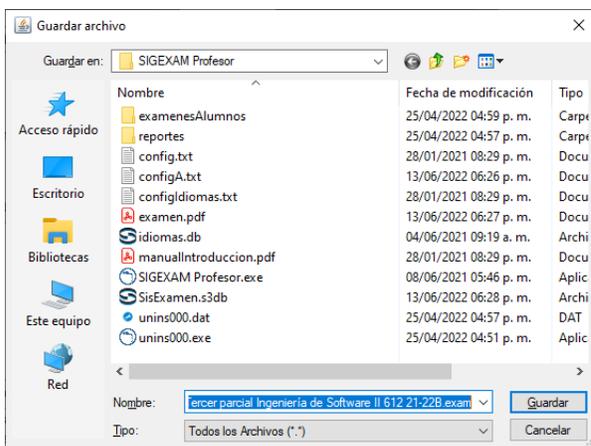


Figura 7. Subsistema profesor. Ventana guardar examen.



Figura 8. Subsistema alumno. Ventana para cargar el examen.

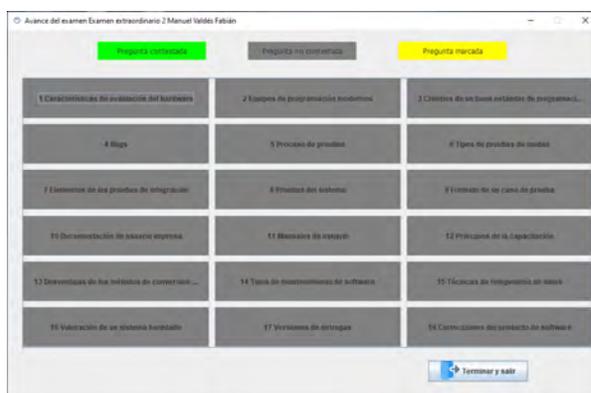


Figura 9. Subsistema alumno. Ventana Avance del examen.

El sistema presenta la ventana Avance del examen para contestar el examen, que es la ventana principal donde el estudiante visualiza el examen completo (Fig.

9). Dentro de esta ventana se encuentran botones que corresponden a las preguntas del examen. En la parte superior de la ventana se muestra el código de colores para las preguntas: el color gris indica preguntas sin responder o respondidas parcialmente (por ejemplo, en una pregunta OM con dos respuestas, tener marcadas cero, uno, o incluso tres o más respuestas), el color verde indica preguntas respondidas en su totalidad (por ejemplo, en una pregunta OM con dos respuestas, tener marcadas solo dos respuestas), y el color amarillo indica preguntas marcadas por el propio estudiante para revisar o corregir más adelante.

El estudiante puede navegar libremente por las preguntas del examen, accediendo a la pregunta que desee las veces que sean necesarias con la posibilidad de corregir las respuestas, como se haría en la mayoría de los exámenes en papel. Cada que se realiza un cambio, se guarda el avance. Las preguntas se visualizan según su tipo en la ventana Presentar examen, como se mostró en las Figuras 2, 3 y 4. El estudiante puede marcar una pregunta para dejarla pendiente y recordar que no se contestó o para revisarla luego porque se tiene duda en la respuesta. Puede quitar las marcas si lo desea, pero no es necesario. Dichas marcas no afectan la revisión del examen.

Cuando todas las preguntas hayan sido contestadas o cuando el estudiante decida terminar el examen, se debe usar la opción de finalizar el examen. Esto realizará el proceso automatizado de calificado, y al finalizar abrirá la ventana Detalles del examen que muestra los resultados del examen (Fig. 10). Esta ventana utiliza un código de colores para indicar los resultados de cada pregunta: color verde para preguntas contestadas correctamente, color rojo para preguntas incorrectas y color amarillo para preguntas

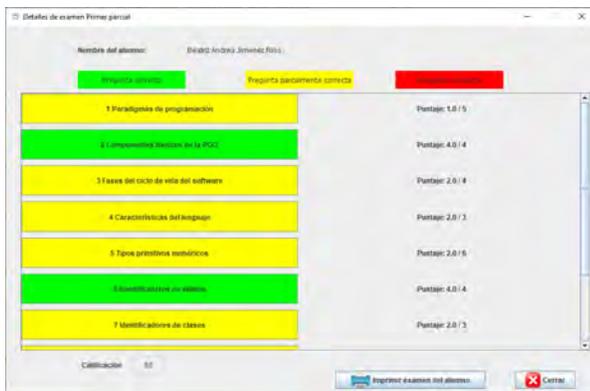


Figura 10. Subsistema alumno. Ventana Detalles del examen.

parcialmente correctas. También, se muestra la puntuación obtenida en cada una de las preguntas. Además, el estudiante puede visualizar su examen contestado en PDF. Los estudiantes pueden consultar las preguntas del examen contestado seleccionando la pregunta deseada y recorriendo sus respuestas, con una versión simplificada de la ventana Presentar examen, llamada Mostrar examen (Fig. 11). En esta ventana no se indica cuáles son las respuestas correctas o incorrectas, tampoco permite modificarlas, para evitar posibles trampas.

Un examen contestado ya no podrá ser contestado de nuevo, solo visualizado directamente desde la ventana Detalles del examen. Las respuestas se guardan en el mismo archivo EXAM abierto originalmente, el cual se renombra agregando el nombre del estudiante que contestó el examen. Este archivo es el necesario para compartir con el docente para entregar sus respuestas. El docente recibe los archivos de exámenes de los estudiantes y los carga en el subsistema profesor (Fig. 12). Una vez cargados podrá ver la calificación y el detalle del examen, podrá imprimirlos o enviar los archivos a resguardo.

SIGEXAM como desarrollo de software
SIGEXAM puede servir como ejemplo de



Figura 11. Subsistema alumno. Ventana Mostrar examen.

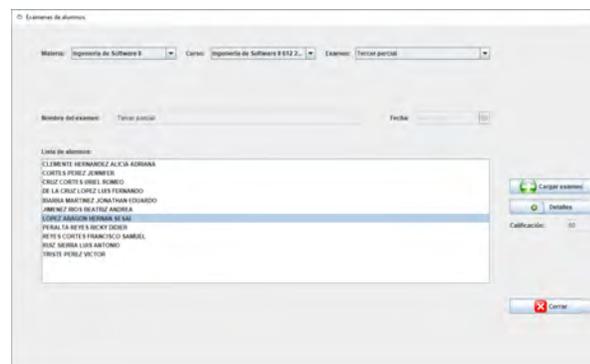


Figura 12. Subsistema profesor. Exámenes cargados.

desarrollo de un sistema para otros desarrolladores y estudiantes, ya que cuenta con la debida documentación de un desarrollo de *software*. El modelo de proceso de *software* utilizado para desarrollar el proyecto es el modelo de la cascada (Fig. 13). Este modelo, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del *software*, que comienza con la especificación de los requerimientos por parte del cliente, avanza a través de las etapas de planeación, modelado y construcción, hasta finalizar con el despliegue del *software* terminado. El modelo de la cascada sirve como un modelo de proceso útil en situaciones en las que los requerimientos son fijos y el trabajo avanza en forma lineal hacia el final (Pressman 2010). Dado

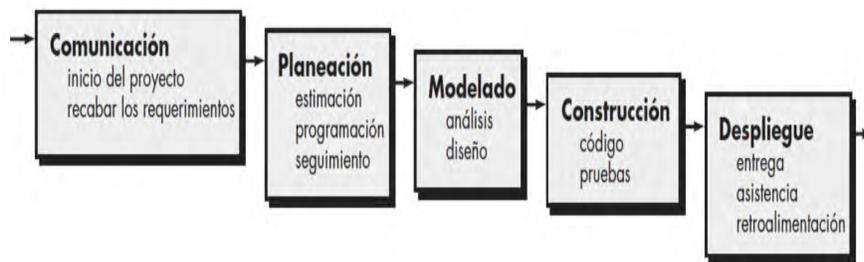


Figura 13. Modelo de la cascada.

que estas situaciones fueron ciertas para el proyecto, se utilizó este modelo para desarrollar el producto de *software* SIGEXAM. A continuación, se resumen las cinco etapas en las que consistió el proyecto de desarrollo de SIGEXAM, más la puesta en producción de este.

1) Comunicación del sistema

Para esta etapa, se identificaron los requerimientos funcionales de cada subsistema y se realizó un listado de requerimientos como producto. En dicho documento se desglosan en cinco módulos los requerimientos funcionales del subsistema profesor (pantalla principal, profesor, materias, cursos y ayuda) y en cuatro módulos los del subsistema alumno (registro de alumno y búsqueda de examen, avance del examen, presentar examen y detalles del examen).

2) Planeación del sistema

El desarrollo del sistema estuvo sujeto a la disponibilidad de recursos humanos, en forma de estancias académicas, estancias profesionales y servicios sociales; las estancias con una duración de siete semanas exactas y los servicios sociales con una duración de mínimo seis meses. El esfuerzo real fue menor a dos años; pero el tiempo del proyecto fue mayor, ya que el desarrollo no fue continuo. Las claves para el seguimiento exitoso del proyecto fue el uso de estándares de programación y la disciplina de documentar el código fuente, ya que los programadores

estuvieron en el proyecto en diferentes periodos y no interactuaron directamente.

3) Modelado del sistema

Para el modelado, primero se diseñó la interfaz gráfica de usuario (GUI) de ambos subsistemas, empleando la metodología de prototipos desechables utilizando el *software* *Pencil*. Se realizaron prototipos para los dos subsistemas. Cabe mencionar que el prototipo del subsistema del alumno está basado en el examen de certificación de Java, presentado por Sierra & Bates (2017). A la par de los prototipos, se diseñó la base de datos para el subsistema profesor y la base de datos para los exámenes individuales, además de una base de datos para el manejo de idiomas. Se generó un modelo entidad-relación (Fig. 14) y un diccionario de datos por cada base de datos. Dado que el sistema es orientado a objetos, se tuvieron que identificar las clases del dominio del problema y la solución. Estos diagramas van evolucionando conforme se va avanzando en el diseño y en la codificación. Para efectos de documentación, la versión definitiva de los diagramas de clases se obtuvo mediante ingeniería inversa utilizando el *plug-in* de Eclipse ObjectAid UML (Fig. 15).

4) Construcción del sistema.

Para codificar el *software* SIGEXAM se utilizó el lenguaje de programación orientado a objetos Java en su versión 8, con el entorno de desarrollo Eclipse IDE en diferentes versiones. La documentación

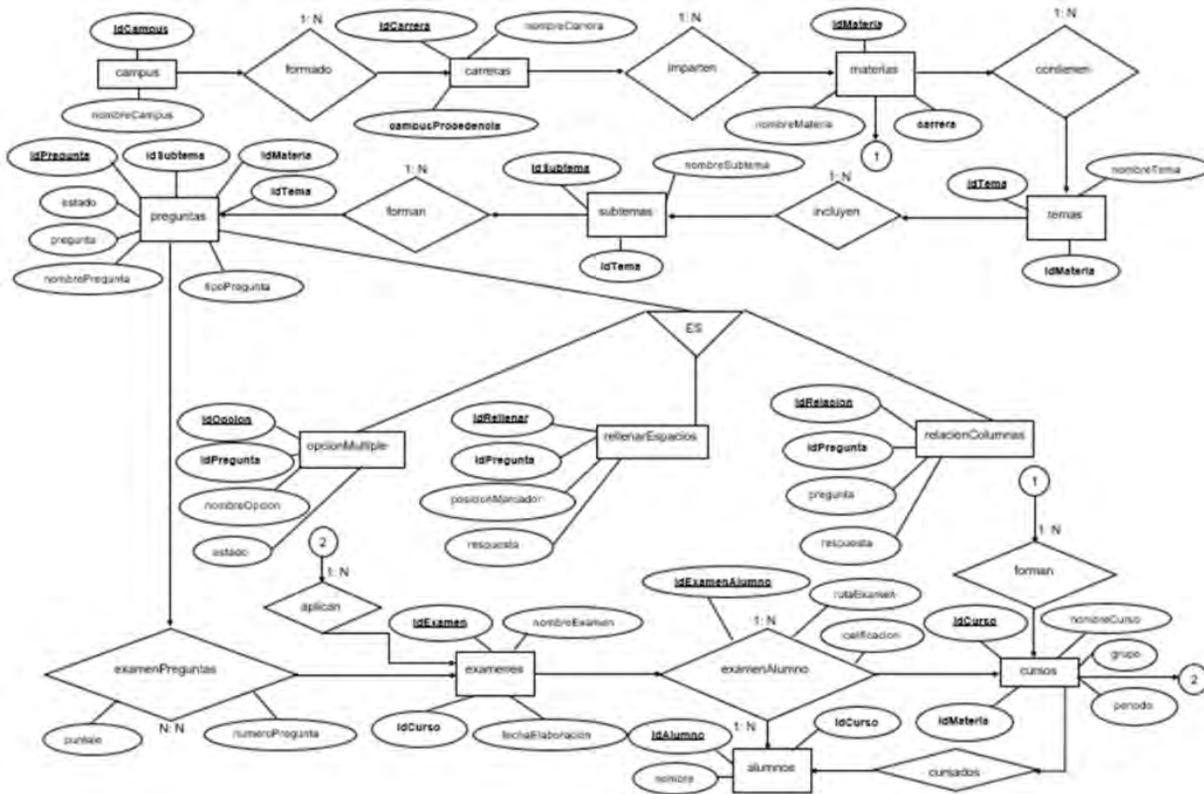


Figura 14. Subsistema profesor. Modelo entidad-relación. Diseño conceptual.

del código fuente fue realizada a través de comentarios Javadoc, los cuales fueron escritos por cada programador. Las bases de datos se implementaron en SQLite. En todo el sistema, se cuidó la integridad referencial de los datos; por ejemplo: no se puede modificar o eliminar una pregunta del sistema si ya fue utilizada en algún examen, no se puede modificar o eliminar un examen del sistema cuando ya se tienen exámenes contestados por los alumnos. El subsistema del profesor consta de 37 clases divididas en 6 paquetes, con casi 19,000 líneas de código. El subsistema del alumno consta de 21 clases divididas en 5 paquetes, con más de 7,500 líneas de código.

SIGEXAM ha sido probado y depurado en cada etapa de su desarrollo e incluso ha sido utilizado en exámenes reales desde que estaba en fase beta. Durante la etapa de pruebas, dentro de esta misma

fase de construcción, se realizaron pruebas de unidad, de integración, de sistemas y de aceptación. Cuando alguna prueba cumplió su objetivo –recordando que las pruebas buscan revelar la presencia de defectos, pero no sirven para demostrar la ausencia de ellos (Dijkstra 1972, citado por Schach 2006)– se depuraron los errores. Es importante mencionar que durante la integración se ejecutaron pruebas de humo cada vez que se realizó un cambio en alguno de los subsistemas, para evitar dañar partes ya funcionales. Las pruebas de aceptación fueron hechas en diversas evaluaciones con un grupo de la Licenciatura en Informática en la materia de Metodología de la Investigación. Los exámenes se aplicaron de forma presencial, fueron creados bajo la modalidad manual con los tres tipos de preguntas. Estas pruebas fueron realizadas en un ambiente no controlado, ya que se utilizaron las computadoras personales de los

alumnos. El intercambio de archivos fue con memorias USB, demostrando así que no se necesita una red local o servicio de internet para utilizar el sistema.

5) Despliegue del sistema

Dentro de la última fase del modelo de la cascada, se llevó a cabo la entrega del producto de software, esto es: los ejecutables de los dos subsistemas, ambos con su documentación de usuario. Para que un *software* sea considerado un verdadero producto de calidad, aparte del programa de cómputo debe tener sus datos asociados y su documentación de usuario (Sommerville 2011). Se elaboró un instalador y un manual de instalación para cada subsistema. También, se elaboró un manual de introducción para cada subsistema. Los manuales de introducción en formato electrónico están incorporados a SIGEXAM y desde el sistema se pueden consultar. Para elaborar los instaladores, se utilizaron varias herramientas: Eclipse IDE para generar el ejecutable Java de la aplicación, Launch4j para crear el ejecutable Windows a partir del ejecutable Java, Inno Setup Compiler para crear el programa de instalación en Windows a partir del ejecutable, y por último AutoPlay Media Studio para crear el disco de instalación a partir del programa de instalación y los complementos necesarios para ejecutar el sistema.

El estándar IEEE 1063-2001 se utilizó para elaborar la documentación de usuario. Este estándar proporciona requisitos mínimos para la estructura, el contenido de la información y el formato de la documentación del usuario, incluidos documentos impresos y electrónicos utilizados en el entorno de trabajo por los usuarios de sistemas que contienen *software*. Este estándar se aplica a los manuales de usuario impresos, ayuda en línea y documentación de referencia del usuario (IEEE 2001).

El estándar recomienda que la documentación que se brinda por separado (en este caso, el manual de instalación) incluya los requisitos de *hardware* y *software* del sistema, información sobre la instalación, recuperación y solución de problemas que surgen durante la instalación, instrucciones de cómo iniciar el *software*, instrucciones para acceder a la documentación electrónica e información de contacto para soporte técnico. Se siguieron todas las recomendaciones. En cuanto a la documentación para usar el sistema, el estándar ofrece dos modalidades: modo instrucción, para enseñar a los usuarios menos experimentados a utilizar el sistema, con base en ejemplos o ilustraciones; y modo referencia, para ofrecer a usuarios familiarizados con el *software* el acceso directo a información específica sobre las funcionalidades del sistema. Se optó por modo instrucción. El estándar recomienda que la documentación de usuario con modo instrucción incluya índices, información para el uso de la documentación, procedimientos, mensajes de error y su solución, así como la navegación entre pantallas. Los procedimientos incluyen información necesaria para iniciar el procedimiento, el paso a paso del proceso, y la información para completar cada tarea.

Adicionalmente, la planificación inicial del proyecto contemplaba pruebas piloto presenciales en un entorno real para tener opiniones adicionales a las de los propios programadores del sistema. Una prueba piloto es un sistema de prueba implementado en un subconjunto de la operación total, como una oficina o un área geográfica; si la prueba tiene éxito, se puede implementar en el resto de la operación (McLeod 2000). Las pruebas serían en salas de cómputo, ya que cada alumno necesita estar en una computadora, y con grupos distintos a Informática, para

probar la facilidad de uso del sistema con estudiantes no tan acostumbrados a utilizar *software*. Sin embargo, todas las actividades presenciales fueron canceladas por la pandemia de COVID-19, y dado que todos los estudiantes de dichos grupos no contaban con computadoras personales, se decidió invitar a otros miembros del cuerpo académico de Informática a participar en dicha prueba, con grupos y exámenes reales de evaluaciones parciales. Dichos grupos cumplían con la condición de contar todos con computadoras. A continuación, se mencionan las pruebas que se realizaron y se muestra parte de la evidencia entregada por los profesores que realizaron las pruebas:

Materia Tecnologías Web I. En esta prueba se aplicó una evaluación parcial al grupo de sexto semestre de la Licenciatura en Informática de la UMAR campus Puerto Escondido. El subsistema alumno fue instalado por los propios alumnos en sus computadoras; mientras que el subsistema profesor, en la computadora de profesor. De acuerdo con la evidencia entregada (Fig. 16 y Fig. 17), el profesor realizó lo necesario para elaborar un examen manual con preguntas de tipo “rellenar espacio” y el examen fue gestionado y revisado en SIGEXAM. El intercambio de archivos se realizó con memorias USB.

Materia Computación II. En este caso, se aplicaron los exámenes de una evaluación parcial a un grupo de la Licenciatura en Ciencias Marítimas de la UMAR campus Puerto Ángel. La instalación fue similar al caso anterior. De acuerdo con la evidencia entregada (Fig. 18 y Fig. 19), se hizo un examen manual donde las preguntas fueron del tipo “opción múltiple” y el examen fue gestionado desde la plataforma Google Classroom. En dicha plataforma se publicó el examen y posteriormente los alumnos subieron

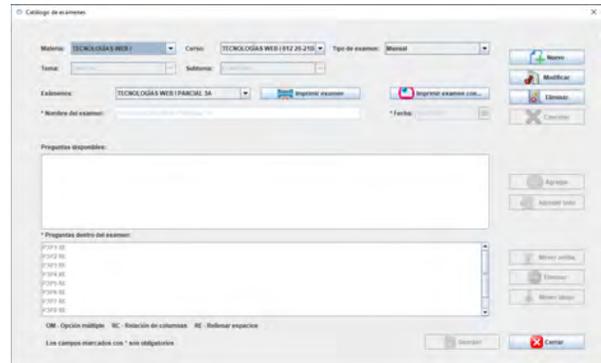


Figura 16. Catálogo de exámenes.



Figura 17. Ventana Detalles de examen.

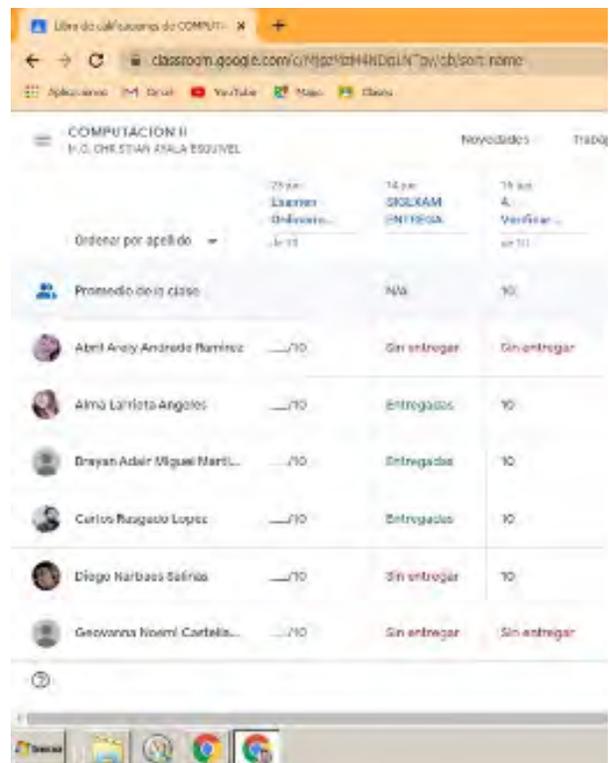


Figura 18. Registro de la actividad examen en Classroom.

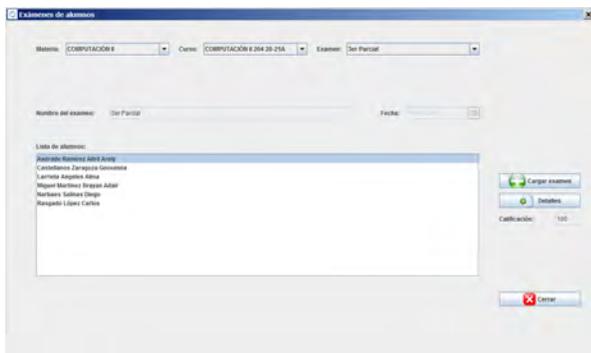


Figura 19. Exámenes de los alumnos ya contestados.

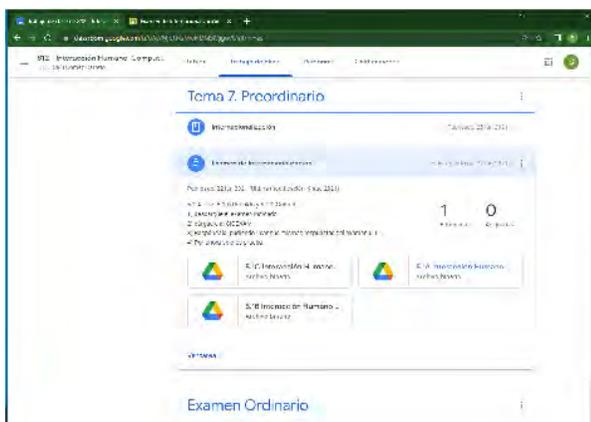


Figura 20. Actividad en Classroom con tres exámenes generados a partir del banco de preguntas.

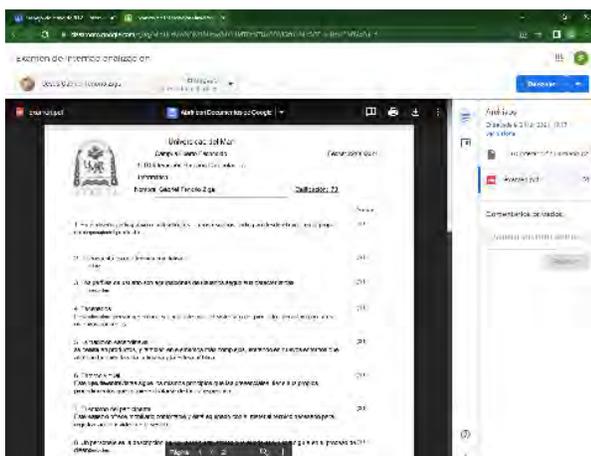


Figura 21. Vista preliminar del examen en PDF desde Classroom.

sus archivos de evidencia del examen contestado.

Materia Interacción Humano-Computadora. En esta prueba se aplicó una evaluación parcial al grupo de octavo semestre de la Licenciatura en Informática de la UMAR campus Puerto Escondido. La instalación fue similar a los casos anteriores. De acuerdo con la evidencia entregada (Fig. 20 y Fig. 21), se hizo un examen automático donde las preguntas fueron de falso y verdadero, lo cual se logra utilizando las preguntas de “opción múltiple” con una respuesta correcta y una incorrecta. El sistema generó con éxito tres exámenes diferentes a partir de un banco de 22 reactivos. Al igual que en el caso anterior, el examen fue gestionado desde Google Classroom con la misma dinámica. En este caso, también se pidió a los estudiantes su opinión respecto al sistema; no hubo comentarios negativos sobre el funcionamiento, aunque sí sobre el diseño de las pantallas, en específico el tamaño de la fuente.

Puesta en producción del sistema

El sistema ha sido utilizado de forma ininterrumpida desde el año 2020 hasta la fecha, tanto en clases a distancia como en presenciales, en diferentes cursos de las materias Ingeniería de *Software* I y II, Paradigmas de Programación I, Metodología de la Investigación y Sistemas de Información I y II de la Licenciatura en Informática, para evaluaciones parciales, ordinarias y extraordinarias. En total se han aplicado 203 exámenes con SIGEXAM hasta julio de 2022.

Cabe destacar que el sistema nunca ha fallado en una calificación automática, aunque obviamente ningún sistema de este tipo está exento de errores humanos al momento de plantear mal una pregunta

o sus respuestas. Tampoco se ha dado el caso que un examen contestado se pierda o que los alumnos copien, claro está, en exámenes con supervisión. Incluso, se tuvo la oportunidad no planeada de demostrar, en un ambiente no simulado, que el sistema es inmune a las fallas en el suministro eléctrico que provocan que se apague el equipo de cómputo en el transcurso del examen; simplemente cuando se restableció el suministro eléctrico, el estudiante volvió a prender su equipo, abrió el examen y continuó contestando sin haber perdido el avance.

Discusión

Entre los hallazgos se confirmó que el lenguaje de programación Java, que fue seleccionado para elaborar el sistema, es realmente independiente de plataforma y por tanto no hubo problemas de compatibilidad al instalar. A pesar de contar con la documentación, muy pocos usuarios necesitaron los manuales para la instalación y el uso de las aplicaciones, lo que se interpreta como que sí se logró que el sistema sea fácil de usar. Los archivos resultantes ocupan poco espacio de almacenamiento. En comparación con las formas de Google, que almacenan el documento en formato PDF, los archivos de SIGEXAM son portables y el sistema puede generar los reportes PDF desde la misma aplicación, con el mínimo espacio de almacenamiento, lo que implica que el sistema es eficiente en el manejo de recursos. Una de las principales fortalezas del SIGEXAM es la seguridad en caso de caída del sistema por falla eléctrica, la aplicación es capaz de retomar el examen en el punto en que se quedó en el momento del percance. Otra fortaleza es la dificultad para copiar en un espacio controlado por el profesor y con exámenes generados aleatoriamente. Dentro de la carrera de Informática, SIGEXAM deja

una evidencia práctica de la aplicación de la Ingeniería de *Software* y el apoyo de diversas herramientas para construir un *software*; un área esencial dentro del ramo profesional requerido en el perfil de egreso. Es importante recalcar que el desarrollo de SIGEXAM se realizó considerando la base reglamentaria, misma que no se cumple en otras aplicaciones.

En las pruebas realizadas, la limitación más importante encontrada por los participantes fue el tamaño de la letra de las pantallas y la sugerencia fue realizar un trabajo de diseño gráfico adicional en el subsistema del alumno a fin de que no se distraiga a los estudiantes, pero que puedan personalizar el tamaño. También, no se tiene una distribución óptima del texto en los exámenes en PDF. Dentro del sistema se tiene limitantes en cuanto a la longitud del texto de las preguntas y respuestas, así como la cantidad de respuestas posibles, debido a la visualización de las preguntas. Otras sugerencias son que en un futuro podrían agregarse otros tipos de preguntas e incluir algún apoyo para discapacidad visual.

Conclusión

SIGEXAM es el resultado de un proyecto de desarrollo de *software* a la medida, ya que se ha tomado como base el Reglamento de Alumnos de la UMAR. Está disponible sin costo alguno para el personal docente de la UMAR que necesite un sistema para gestionar exámenes. También se podría adaptar para ser utilizado en otros centros educativos para evaluar en el aula, hacer evaluaciones diagnósticas o incluso evaluaciones certificativas. Utilizando otros recursos, incluso se puede utilizar para exámenes en línea. Basogain-Urrutia (2021) propone que estos recursos sean:

- Conexión en directo mediante

videoconferencia. Para que el profesor monitoree a los alumnos (y su entorno) mientras aplican el examen, exigiendo cámara y micrófono encendido.

- Programas de bloqueo de escritorio. Para que el alumno no pueda abrir ni ver ninguna otra aplicación excepto la del examen, ni copiar o pegar texto.

- Programas de monitorización y supervisión en vivo (*e-proctoring*). Para monitorear en vivo la actividad en la computadora, así como capturar imágenes y sonido del entorno donde se desarrolla el examen, con funciones de inteligencia artificial para reconocer al estudiante o detectar otras personas.

- Combinación de los anteriores. En caso de considerarse insuficiente solo uno de los anteriores.

Para la construcción del sistema se emplearon diversos estándares de desarrollo de *software*, que permitan una correcta interpretación del problema y llegar a implementar la correcta solución de este en un sistema de información. Se emplearon las herramientas necesarias para cumplir con cada etapa planificada y, aunque se probó en ambientes distintos a los planeados porque las condiciones de pandemia no lo permitieron, finalmente se obtuvieron resultados satisfactorios y se tiene disponible un sistema gestor de exámenes totalmente funcional, que ha servido para aplicar más de 200 exámenes cerrados.

Los trabajos futuros son agregar otros tipos de preguntas cerradas, como multi-reativos (Ceneval 2021) basados en un texto o una imagen; incluir algún apoyo para discapacidad visual, como una herramienta para personalizar la fuente o una herramienta que lea cada pregunta y respuesta; mejorar la visualización de preguntas y la generación de exámenes

en PDF; e incorporar mecanismos de importación/exportación de preguntas, para el intercambio de información entre profesores.

Agradecimientos

A los alumnos del programa “Delfín” Humberto Medina Hernández, Luis Enrique López Hernández y Gabriela Santiago García, a los alumnos de estancias profesionales Cruz Violeta Bautista Juárez y Valentino Velázquez Olivera, y a los alumnos de servicio social Kevin Horacio Reyes de Esesarte e Imelda Perales Ambrocio que apoyaron en diversas etapas del desarrollo. A los profesores-investigadores del Cuerpo Académico de Informática (UMAR-CA-11) que contribuyeron a las pruebas piloto de los subsistemas en diferentes ambientes. A los participantes del proyecto interno 2II2004. A dos revisores anónimos por sus valiosas observaciones.

Referencias

- Basogain-Urrutia, J.X. 2021.** Evaluación en línea: herramientas, limitaciones y alternativas en un contexto de pandemia. *Revista Tecnológica-Educativa Docentes 2.0* 10(2): 30-41. DOI: 10.37843/rted.v10i2.243.
- Bowdish, L. 2016.** Trash to treasure: changing waste streams to profit streams. U.S. Chamber of Commerce Foundation: Corporate Citizenship Center. Consultado el 15 de septiembre de 2022: www.uschamberfoundation.org/sites/default/files/USCCF_TrashToTreasure_v4.pdf
- Ceneval. 2021.** Guía para el sustentante: Examen General para el Egreso de la Licenciatura en Informática (EGEL Plus INFO). Consultado el 8 de noviembre de 2022: www.ceneval.edu.mx/wp-content/uploads/2021/07/EX_GUIA_EGEL_PLUS_INFO.pdf
- IEEE. 2001.** IEEE Standard for Software User Documentation. IEEE Std 1063-2001, Institute of Electrical and Electronics Engineers, Estados Unidos.

- INEE. 2019.** Criterios técnicos y de procedimiento para el análisis de los instrumentos de evaluación. Instituto Nacional para la Evaluación de la Educación. Consultado el 7 de noviembre de 2022: www.dof.gob.mx/nota_detalle.php?codigo=5556441
- INEE. 2018.** Plan Nacional para la Evaluación de los Aprendizajes (PLANEA). Instituto Nacional para la Evaluación de la Educación. Consultado el 1 de septiembre de 2022: www.inee.edu.mx/wp-content/uploads/2019/08/PLANEA_Doctor.pdf
- McLeod, R.Jr. 2000.** Sistemas de información gerencial. 7a ed., Prentice Hall, México, 688 pp.
- Otero-Doval, H. 2014.** El examen, herramienta fundamental para la evaluación certificativa. Pp: 553-562. En: Contreras-Izquierdo, N. M. (ed.), La enseñanza del Español como LE/L2 en el siglo XXI. Asociación para la Enseñanza del Español como Lengua Extranjera - ASELE, España.
- Pressman, R.S. 2010.** Ingeniería del software: un enfoque práctico. 7a ed., McGraw-Hill, México, 777 pp.
- Prometric. 2021.** A medida que las pruebas basadas en computadora continúan creciendo, también lo hacen la seguridad y la integridad de los exámenes. Consultado el 16 de agosto de 2022: www.prometric.com/es/test-owners/resources/medida-que-las-pruebas-basadas-en-computadora-continuan-creciendo-tambien-lo
- Schach, R.S. 2006.** Ingeniería de software clásica y orientada a objetos. 6a ed., McGraw-Hill, México, 581 pp.
- Sierra, K. & B. Bates. 2017.** OCA Java SE 8 Programmer I Exam Guide. Oracle Press, Estados Unidos, 427 pp.
- Sommerville, I. 2011.** Ingeniería de software. 9a ed., Pearson Educación, México, 792 pp.
- SUNEO. 2016.** Reglamento de Alumnos de Licenciatura. Universidad del Mar. Consultado el 9 de septiembre de 2022: www.umar.mx/DocsUMAR/REGLAMENTO_ALUMNOS_UMAR_SUNEO2016.pdf
- Watson, W.R. & S.L. Watson. 2007.** An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become? TechTrends 51(2): 28-34. DOI: 10.1007/s11528-007-0023-y.