

TECNOLOGÍAS EMERGENTES

Un compendio de desarrollos tecnológicos

María del Consuelo Cortés Velázquez
Aurelio Enrique López Barrón
Yobani Martínez Ramírez
Coordinadores

Tecnologías emergentes

Un compendio de desarrollos tecnológicos

Tecnologías emergentes

Un compendio de desarrollos tecnológicos

MARÍA DEL CONSUELO CORTÉS VELÁZQUEZ
AURELIO ENRIQUE LÓPEZ BARRÓN
YOBANI MARTÍNEZ RAMÍREZ
COORDINADORES



UNIVERSIDAD DE GUADALAJARA

Para garantizar la calidad, pertinencia académica y científica de esta obra, el manuscrito fue sometido a un riguroso arbitraje por medio de dictaminado a doble ciego, emitido por académicos especialistas en la materia, avalados por el Comité Editorial del Centro Universitario de la Costa de la Universidad de Guadalajara, México.

La publicación de este libro se financió con recursos del PROFEXCE 2020.

Primera edición, 2020

D.R. © 2020, Universidad de Guadalajara
Centro Universitario de la Costa
Av. Universidad 203
Delegación Ixtapa 48280
Puerto Vallarta, Jalisco, México

ISBN: 978-607-547-991-0

Editado y hecho en México
Edited and made in Mexico

CONTENIDO

Introducción	7
1. Asistente inteligente de voz como apoyo en la educación: material didáctico para prácticas de configuración de equipos de redes y telecomunicaciones	9
Ana Laura Ramírez Vargas	
María del Consuelo Cortés Velázquez	
Aurelio Enrique López Barrón	
2. Sistema de información web para el registro y seguimiento a emprendedores de una incubadora de negocios	41
Julio Francisco Bahena Ayala	
Aurelio Enrique López Barrón	
María del Consuelo Cortés Velázquez	
Karina Donlucas Magaña	
3. Aplicación móvil y sistema de soporte a la decisión para la optimización de solución de reportes en una facultad universitaria	79
Yobani Martínez Ramírez	
Alan Ramírez Noriega	
José Cota Cota	
Martín Sánchez Mundo	
4. Repositorio digital y clasificación automática de tesis.	119
Yobani Martínez Ramírez	
Alan Ramírez Noriega	
Ana María Hernández Ruiz	
María del Carmen Rodríguez Apodaca	

INTRODUCCIÓN

Hoy en día las tecnologías emergentes han abierto un sinnúmero de posibilidades para ser implementadas en diferentes ámbitos de la vida. Desde esta perspectiva, las tecnologías permiten ser un medio para ayudar en el desarrollo de innovaciones que permiten fortalecer procesos y toma de decisiones en las organizaciones. Asimismo, gracias a la generación de nuevas tecnologías permite que muchas de estas organizaciones incrementen sus proyectos con la implementación de las nuevas tecnologías para adaptarlas a sus procesos con el propósito de innovar o mejorar sus productos o servicios.

En la presente obra, el lector podrá encontrar desarrollos tecnológicos en el ámbito educativo y en el ámbito de las organizaciones. Cabe mencionar que los trabajos presentados son productos de proyectos de investigación desarrollados por el Cuerpo Académico 884 “Educación y Desarrollo Tecnológico” adscrito a la Universidad de Guadalajara y el Cuerpo Académico-295 “Sistemas Innova-

dores Aplicados al Contexto Educativo” adscrito a la Universidad Autónoma de Sinaloa, México.

Uno de los temas de desarrollo tecnológico es la implementación de los asistentes inteligentes en el ámbito educativo. En esta temática, se demuestra como las tecnologías emergentes ayudan a innovar para ser utilizados como material didáctico y con ello motivar al aprendizaje de los alumnos. Otro de los temas de desarrollo tecnológico es el diseño e implementación de un sistema de información que tiene el objetivo de efficientar procesos y servicios en una organización. Lo anterior, permite que el uso de este sistema le brinde una ventaja competitiva a la organización.

De manera similar, otro de los temas de desarrollo tecnológico tiene relación con la Internet de las Cosas, en la cual, gracias al diseño e implementación de una aplicación móvil y a un sistema web, se pueden comunicar los problemas y las situaciones que suceden al personal administrativo de una institución educativa. Adicionalmente, permite tanto la gestión de los problemas administrativos como la explotación de la información generada para la toma de decisiones eficaces. Finalmente, el último tema de desarrollo que se presenta en esta obra describe el diseño, desarrollo e implementación de una plataforma web para el repositorio de tesis digital para el uso de una comunidad universitaria.



ASISTENTE INTELIGENTE DE VOZ COMO APOYO EN LA EDUCACIÓN: MATERIAL DIDÁCTICO PARA PRÁCTICAS DE CONFIGURACIÓN DE EQUIPOS DE REDES Y TELECOMUNICACIONES*

ANA LAURA RAMÍREZ VARGAS¹

MARÍA DEL CONSUELO CORTÉS VELÁZQUEZ²

AURELIO ENRIQUE LÓPEZ BARRÓN³

Resumen

El presente documento es una investigación aplicada en la Ingeniería en Telemática (ITEL) del Centro Universitario de la Costa de la Universidad de Guadalajara, México, y tuvo como finalidad desarrollar un *skill* del asistente de voz de Alexa de Amazon como material didáctico de apoyo para las prácticas de configuración de equipos de redes y telecomunicaciones. Dichas configuraciones son

* Derivado del proyecto de investigación: "Actividades de Investigación del Cuerpo Académico UDG-CA-884".

1 Universidad de Guadalajara, correo electrónico: ing.analaura.rv@gmail.com

2 Universidad de Guadalajara, correo electrónico: maria.cortes@academicos.udg.mx

3 Universidad de Guadalajara, correo electrónico: aurelio.lbarrón@academicos.udg.mx

realizadas por los alumnos de la ITEL del primer al octavo semestre en las unidades de aprendizaje correspondientes al eje de conocimientos de redes. Los asistentes virtuales son robots que responden a comandos de voz; cada vez más sofisticados e inteligentes, estos sistemas aprenden a través de interacciones con el consumidor y mejoran continuamente su repertorio: entre más interactúan con las personas más aprenden (Valois, 2018). Por lo anterior, para el diseño y desarrollo de la *skill* se tomó como referencia el modelo de “Ciclo de vida de cascada” propuesto por Pressman (2010). Dicho modelo está diseñado en cinco etapas: a) análisis, b) diseño, c) codificación, d) pruebas, y e) mantenimiento. Finalmente, con el desarrollo e implementación de este material didáctico se demostró la funcionalidad para fortalecer el proceso de enseñanza-aprendizaje como un recurso innovador diseñado con apoyo de las tecnologías emergentes como son los sistemas inteligentes.

Palabras clave: asistentes virtuales, material didáctico, sistemas inteligentes.

Abstract

This document is an applied research in the Telematics Engineering (ITEL) of the University Center of the Coast of the University of Guadalajara, Mexico, and its purpose was to develop an Amazon Alexa voice assistant skill as support didactic material for the network and telecommunications equipment configuration practices. These configurations are made by ITEL students from the first to the eighth semester in the learning units corresponding to the axis of knowledge of networks. Virtual assistants are robots that respond to voice commands; increasingly sophisticated and intelligent, these systems learn through interactions with the consumer, and continually improve their repertoire; the more they interact with people, the more they learn (Valois, 2018). Therefore, for the design and development of the skill, the model “Cascade life cycle” proposed by Pressman

(2010) was taken as a reference. This model is designed in five stages: a) analysis, b) design, c) coding, d) tests, and e) maintenance. Finally, with the development and implementation of this didactic material, the functionality to strengthen the teaching-learning process was demonstrated as an innovative resource designed with the support of emerging technologies such as intelligent systems.

Introducción

Desde el año 1975, la universidad donde se desarrolló este proyecto de investigación se ha distinguido como una de las más importantes de México, tanto por sus actividades de enseñanza, investigación, difusión de la ciencia y la cultura, así como también por su tamaño y cobertura. Con poco más de 200 mil alumnos de nivel técnico, bachillerato, licenciatura y posgrado, es la institución educativa pública más importante del occidente del país (Universidad de Guadalajara, 2019).

Tras un proceso de redefinición y reestructuración institucional, a principios de la década de los noventa se creó en esta universidad una estructura de red que permitió la creación de seis centros universitarios en la ciudad de Guadalajara, organizados por áreas de conocimientos, y de ocho centros universitarios regionales, con oferta multidisciplinaria, distribuidos en las principales zonas geográficas del estado de Jalisco. Con esta medida, se pudo llevar la educación ofertada por la universidad a las comunidades en las que anteriormente los padres tenían que enviar a sus hijos a la ciudad de Guadalajara para poder proporcionar una formación a nivel profesional (Universidad de Guadalajara, 2019).

Una de las regiones a las cuales la Universidad de Guadalajara llevó su oferta educativa es a la zona costa norte del estado de Jalisco, cercana a la zona sur del estado de Nayarit. La ciudad de Puerto Vallarta es la localidad principal de esta zona, por lo cual en el año de 1994 se creó el Centro Universitario de la Costa (CUCosta) en la localidad de Ixtapa, con la finalidad de cubrir la demanda educativa

de una población de aproximadamente 151 mil habitantes (Centro Universitario de la Costa, 2019a).

Actualmente, en 2020 se cuenta con una matrícula de 6,123 alumnos correspondientes a 20 programas de nivel licenciatura en las tres áreas: a) ingenierías, b) ciencias biológicas y de la salud, y c) división de socioeconómicos. Adicionalmente, se cuenta con 28 diplomados en diferentes áreas, siete maestrías y dos doctorados (Centro Universitario de la Costa, 2020c).

En este contexto, el objeto de este proyecto de investigación se enfoca en el área de Ingenierías, más en específico, la Ingeniería en Telemática (en adelante ITEL). Dicha carrera se dictaminó con el propósito de atender las necesidades de tener profesionistas especializados en el campo de las telecomunicaciones. Para respaldar la formación especializada de ITEL, los directivos de la carrera tienen vigente un convenio con la empresa Cisco Systems con la finalidad de incluir al plan curricular algunas materias del Programa Cisco Networking Academy. Este programa es definido como *e-learning* y enseña a los alumnos las habilidades tecnológicas de Internet esenciales en una economía global. Es un programa instruccional que proporciona contenido basado en la Web, pruebas en línea, seguimiento del desempeño de los estudiantes, laboratorios en vivo, soporte técnico y capacitación a profesores para las certificaciones estándar de la industria (Cisco Networking Academy, 2020). Actualmente, el programa educativo de la ITEL cuenta con un plan de estudios diseñado conforme las nuevas tecnologías, el cual tiene como misión la siguiente:

Contribuir a la formación y desarrollo del conocimiento mediante el compromiso y preparación de Ingenieros en Telemática transmitiendo capacidades y responsabilidad para la competitividad de profesionales capaces de enfrentar los nuevos avances tecnológicos de todo el mundo (Centro Universitario de la Costa, 2020c).

En el plan de estudios de ITEL se tienen registradas un total de 61 unidades de aprendizaje y 393 créditos, de las cuales nueve corresponden

al eje del área de conocimiento de redes, siete al eje de conocimientos de electrónica, ocho al eje de conocimientos de matemáticas, siete al eje de conocimientos de informática, tres al eje de conocimientos de *software* de base, seis al eje de conocimientos de entorno social y dos al eje de conocimientos de interfaz hombre-máquina. Adicionalmente, se tienen contempladas seis unidades de aprendizaje del área optativa abierta y doce unidades de aprendizaje correspondientes al área de especialización divididas en las siguientes cuatro áreas de conocimiento: a) redes convergentes, b) seguridad, c) redes inalámbricas, y d) sistemas embebidos (“Dictamen de Creación del Plan de Estudios de la Licenciatura en Ingeniería en Telemática”, 2017).

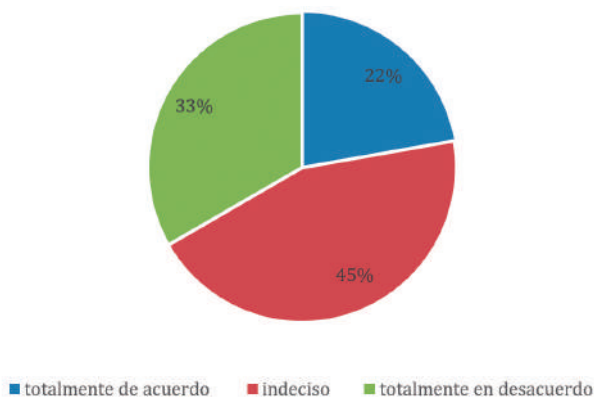
En el caso del eje del área de conocimientos de redes se contemplan las siguientes unidades de aprendizaje: a) introducción a las redes, b) cableado estructurado, c) conmutación y enrutamiento, d) escalabilidad de redes, e) conexión de redes, f) redes emergentes e inalámbricas, g) comunicaciones móviles, h) administración y diseño de redes, e i) seguridad en redes. Dichas unidades de aprendizaje se ofertan desde el primer semestre al octavo semestre de la carrera de ITEL. En cada una de estas unidades de aprendizaje se realizan prácticas de configuración con equipos de redes y telecomunicaciones, como son *routers* y *switchs* con diferentes niveles de complejidad. Dichas prácticas de laboratorio son desarrolladas con base en el contenido del Programa de Cisco Networking Academy, el cual tiene como objetivo promover el desarrollo de destrezas y la formación profesional en tecnologías de la información disponible para entidades educativas y personas en el mundo (Cisco Networking Academy, 2020).

En este contexto, los alumnos de la ITEL enfrentan nuevos desafíos diariamente en su proceso de enseñanza-aprendizaje, al realizar continuamente prácticas de laboratorio en las cuales es necesario configurar los equipos de redes y telecomunicaciones. Dichas prácticas son realizadas de forma física y por medio del simulador llamado “Packet Tracer” (Cortés, comunicación personal, 30 de enero de 2019).

Con base en lo anterior, se diseñó un instrumento el cual tuvo como propósito identificar la percepción de los alumnos de la carrera de ITEL sobre la percepción que tienen en la realización de las prácticas de laboratorio para la configuración de equipos de redes y telecomunicaciones. Dicho instrumento fue un cuestionario nombrado “Percepción de prácticas de Laboratorio” (PPL), el cual fue aplicado al 100 % de los alumnos de la carrera de ITEL, siendo un total de 160 alumnos. Dicho instrumento se conformó de cinco preguntas cerradas. En la figura 1 se muestran los porcentajes que hacen referencia a la pregunta uno, relacionada con la facilidad de recordar los comandos para hacer las prácticas de configuraciones de equipos de redes y telecomunicaciones; quedando de la siguiente manera: 1. “totalmente de acuerdo” 22 %, 2. “indeciso” 45 %, y 3. “totalmente en desacuerdo” 33 %.

Figura 1

Facilidad de recordar los comandos para hacer las configuraciones de equipos de redes y telecomunicaciones

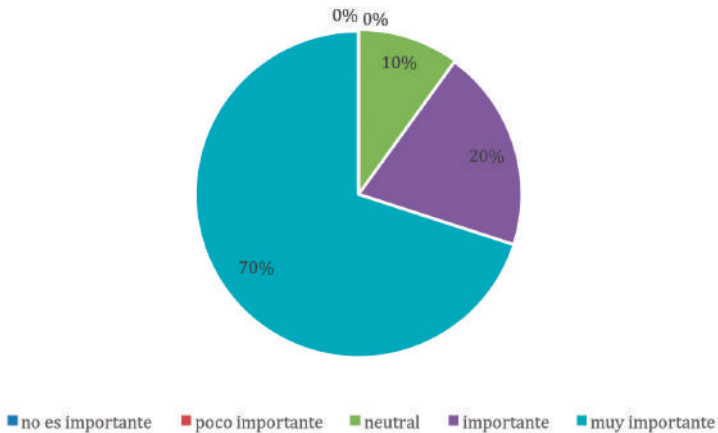


En la figura 2 se muestran los porcentajes que hacen referencia a la pregunta dos, relacionada con la importancia de realizar prácticas de configuración de los equipos de redes y telecomunicaciones para el desarrollo de habilidades en la carrera; quedando de la siguiente

manera: 1. “no es importante” 0 %, 2. “poco importante” 0 %, 3. “neutral” 10 %, 4. “importante” 20 %, y 5. “muy importante” 70 %.

Figura 2

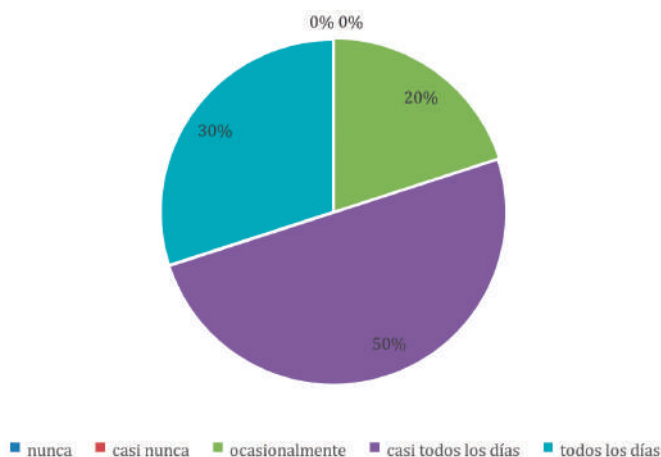
Importancia de realizar prácticas de configuración de los equipos de redes y telecomunicaciones para el desarrollo de habilidades en su carrera



En la figura 3 se muestran los porcentajes que hacen referencia a la pregunta tres, relacionada con la frecuencia de tiempo en usar los comandos para la realización de prácticas de laboratorio de configuración de equipos de redes y telecomunicaciones; quedando de la siguiente manera: 1. “nunca” 0 %, 2. “casi nunca” 0 %, 3. “ocasionalmente” 20 %, 4. “casi todos los días” 50 %, y 5. “todos los días” 30 %.

Figura 3

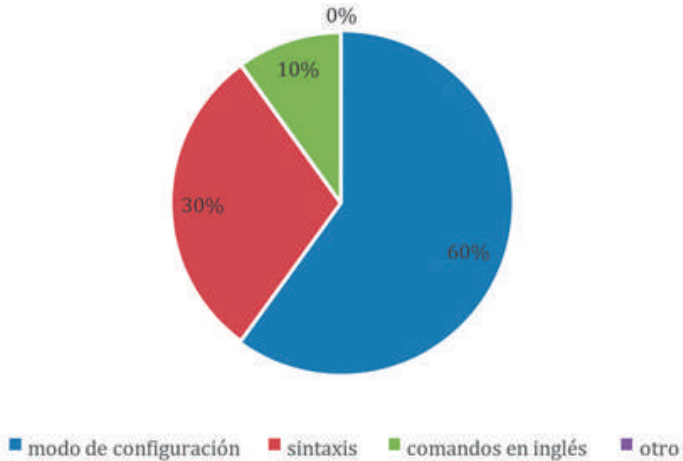
Frecuencia de usar los comandos para la realización de prácticas de laboratorio de configuración de equipos de redes y telecomunicaciones



En la figura 4 se muestran los porcentajes que hacen referencia a la pregunta cuatro, relacionada con las problemáticas que se tienen al realizar las prácticas de laboratorio de configuración de equipos de redes y telecomunicaciones; quedando de la siguiente manera: 1. “me es difícil recordar en qué modo de configuración se utilizan” 60 %, 2. “me es difícil memorizar la sintaxis de los comandos” 30 %, 3. “los comandos están en inglés” 10 %, y 4. “otra” 0 %.

Figura 4

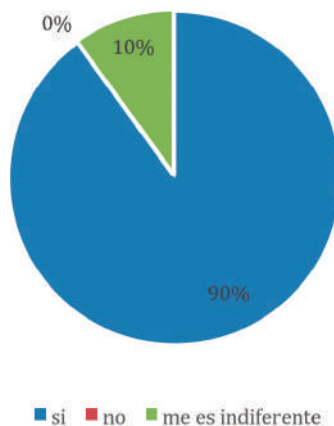
Problemáticas que se tienen al realizar las prácticas de laboratorio de configuración de equipos de redes y telecomunicaciones



En la figura 5 se muestran los porcentajes que hacen referencia a la pregunta cinco, relacionada a si les gustaría apoyarse de un asistente de voz virtual para realizar las configuraciones de los equipos de redes y telecomunicaciones; quedando de la siguiente manera: 1. “sí” 90 %, 2. “no” 0 %, y 3. “me es indiferente” 10 %.

Figura 5

Les gustaría apoyarse de un asistente de voz virtual para realizar las configuraciones de los equipos de redes y telecomunicaciones



Con base en los resultados de la encuesta realizada a los alumnos de la ITEL, se puede concluir lo siguiente: Primero, son minoría los alumnos quienes mencionaron que es fácil recordar los comandos. Segundo, la mayoría de los alumnos expresaron que es muy importante realizar prácticas de configuración de los equipos de redes y telecomunicaciones para el desarrollo de habilidades en su carrera. Tercero, la mayoría de los alumnos manifestaron que casi todos los días usan los comandos para la realización de prácticas de laboratorio de configuración de equipos de redes y telecomunicaciones. Cuarto, los alumnos mencionaron que la principal problemática en realizar las prácticas de laboratorio es que se les dificulta recordar el modo de configuración donde se utilizan los comandos, seguido de la dificultad de memorizar la sintaxis de los comandos. Finalmente, la mayoría de los alumnos mencionaron que les gustaría apoyarse de un asistente de voz virtual para realizar las configuraciones de los equipos de redes y telecomunicaciones. Por lo anterior, el poder contar con un asistente virtual de voz podría ayudar a los alumnos a tener un material didáctico que les sea atractivo para realizar las prácticas de configuración de los equipos.

Por lo anterior, el propósito de este proyecto de investigación es desarrollar una *skill* de asistente inteligente de voz como material didáctico para la configuración de equipos de redes y telecomunicaciones para los alumnos de la carrera de ITEL con la finalidad de apoyar en su proceso de enseñanza-aprendizaje. Asimismo, los objetivos específicos fueron los siguientes: a) identificar los requerimientos del asistente inteligente de voz para la configuración de los equipos de redes y telecomunicaciones; b) construir un diseño para el asistente inteligente de voz, para la configuración de los equipos de redes y telecomunicaciones; c) generar el código de programación del asistente inteligente de voz, para la configuración de los equipos de redes y telecomunicaciones; d) evaluar la funcionalidad del asistente inteligente de voz para la configuración de los equipos de redes y telecomunicaciones; y e) mantener al asistente inteligente de voz para su correcto funcionamiento, para la configuración de los equipos de redes y telecomunicaciones.

En este contexto, el diseñar y desarrollar un asistente virtual para realizar las configuraciones de redes y telecomunicaciones para los alumnos de la ITEL se justificó específicamente por cuatro razones. La primera, porque los alumnos de esta carrera expresaron con base en el cuestionario PPL que se les dificulta aprender los comandos, por lo que el asistente virtual será un material didáctico tecnológico que apoyará en el proceso de enseñanza-aprendizaje. La segunda, los instructores del Programa de Cisco Networking Academy, quienes imparten cursos a los alumnos de la ITEL, mencionaron que el contar con un material educativo tecnológico podría apoyar a favorecer la motivación en los alumnos en aprender los comandos para la realización de las prácticas de los equipos cisco (Cortés y López, comunicación personal, 5 de octubre de 2019).

La tercera, de acuerdo al *Plan de Desarrollo Institucional de la Universidad de Guadalajara 2014-2030* se ha establecido como una de las acciones a realizar la correspondiente a impulsar el uso de las tecnologías de la información y la comunicación (en adelante TIC) en las diversas modalidades de educación para promover el desarrollo

de las capacidades de los estudiantes y fomentar nuevas pedagogías de aprendizaje (Universidad de Guadalajara, 2010). Por lo anterior, el contar con un asistente virtual como material didáctico apoyaría a dar cumplimiento a esta acción.

Finalmente, en la misión de la ITEL se menciona lo siguiente:

Contribuir a la formación y desarrollo del conocimiento mediante el compromiso y preparación de Ingenieros en Telemática transmitiendo capacidades y responsabilidad para la competitividad de profesionales capaces de enfrentar los nuevos avances tecnológicos de todo el mundo (Centro Universitario de la Costa, 2020d).

Por lo anterior, el contar con un asistente virtual como material didáctico para realizar las configuraciones del equipo redes y telecomunicaciones ayudará a contribuir con la misión de la carrera.

Revisión de la literatura

Los materiales educativos son recursos para el aprendizaje, son “todos los medios y los recursos que facilitan el proceso de enseñanza y la construcción de aprendizaje” (López, 2006). De acuerdo con el autor antes mencionado, los materiales educativos buscan estimular las funciones sensoriales, crear experiencias y conocimientos previos con los cuales se pueda acceder más fácilmente a la información necesaria para el desarrollo de capacidades físicas y destrezas, la formación de actitudes y valores. Los materiales educativos presentan características como: a) materiales impresos o no, que facilitan el aprender a aprender, pueden ser impresos, audiovisuales, informáticos y otros; b) están formados por herramientas con un papel didáctico o medio facilitador de información; c) son utilizados como medio para estimular y orientar el proceso educativo; d) permiten a los estudiantes adquirir información, experiencias, desarrollar actitudes y normas de conducta de acuerdo al objetivo a lograr; e) son herramientas que fortalecen las propuestas de enseñanza; y f) suelen

denominarlos materiales educativos como recursos o medios para lograr el aprendizaje.

Por otro lado, el autor Artigas (citado por Simonit, 2009) hizo una distinción entre materiales educativos y materiales didácticos, ya que el material educativo es destinado al docente porque su objetivo es tener claro lo que quiere enseñar y qué hace para llegar al estudiante como intermediario del proceso de aprendizaje. Por otro lado, el material didáctico se considera como el que va directo a las manos de los estudiantes y funciona como mediador instrumental, incluso cuando un adulto no está capacitado para un área de aprendizaje en específico.

Por su parte, Cruzat (2009) mencionó que los materiales educativos son medios que permiten estimular y orientar el proceso educativo, son auxiliares didácticos que dan la oportunidad al docente de manera práctica y objetiva de visualizar resultados satisfactorios de proceso enseñanza-aprendizaje. Entonces, los materiales educativos son todos los elementos, herramientas o recursos que conlleven a la estimulación y orientación del proceso educativo y que permitan obtener experiencias sensoriales que puedan los estudiantes relacionar a un conocimiento en específico. Además de crear una motivación en los estudiantes a aprender más.

En este contexto, de acuerdo con Ramírez (2011), el uso de las TIC está revolucionando la forma de aprendizaje de los alumnos de las nuevas generaciones, y esto provoca que niños y adultos aprendan más del entorno que los rodea con las herramientas que la tecnología provee. La inmersión tecnológica a la educación ha permitido implementar nuevas tecnologías en el aula como plataformas de educación virtual, acceso a documentación y recursos desde la comodidad de su casa o aula, el uso de dispositivos que ayuden a los estudiantes a aprender e incluso que los profesores puedan impartir clases virtualmente. El concepto del *e-learning* ya utilizado desde algún tiempo en los procesos formativos y con acceso al avance tecnológico, las telecomunicaciones y las redes de datos, han permitido el nacimiento del concepto de *mobile learning* o aprendizaje móvil. El aprendizaje apoyado en la tecnología desde cualquier lugar y en cual-

quier momento es la mejor apuesta que puede hacer un docente. Un ejemplo de esto son los materiales didácticos web para dispositivos móviles, el que se pueda tomar como herramienta para aprendizaje un dispositivo, ya sea un celular, una tableta, una *laptop* o cualquier otro dispositivo que tenga conexión a Internet, como una bocina inteligente (Ramírez, 2011). Los materiales educativos a los que se debería apostar en la actualidad son los que fomenten el aprendizaje entre los alumnos de manera autodidacta y que les permitan manejar información a través de dispositivos conectados a Internet y con inteligencia artificial (en adelante IA). De acuerdo con Zampayo (2004), la IA se define de la siguiente manera:

Inteligencia, es la capacidad de comprender, evocar, movilizar e integrar constructivamente lo que se ha aprendido y de utilizarlo para enfrentarse a nuevas situaciones. Artificial, es aquel cuyo producto origen no es natural, sino que fue hecho por la mano o arte del hombre.

Amazon (2000) definió la IA como el campo de la ciencia informática dedicado a la resolución de problemas cognitivos asociados con la inteligencia humana, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. La IA puede provocar escenas futuristas o robóticas, pero va más allá de la ciencia ficción, llegando a lo que hoy va avanzando la ciencia informática. Entonces, la IA se vuelve “más inteligente” y aprende aún más rápido conforme más datos posee, y las empresas van generando más datos con soluciones de Internet de las Cosas, en sus siglas en inglés “IoT”, la tecnología de los sensores, que aporta más datos para analizar, como lugares y eventos.

Según los expertos del informe “Artificial Intelligence and Life in 2030”, las herramientas de inteligencia artificial “van a convertirse en uno de los núcleos del proceso de enseñanza en la educación superior”, ya que ayudarán a mantener los costos al cubrir un mayor número de alumnos y permitir que aumente el rendimiento y aprendizaje escolar (Aika Educación, 2016). En cuanto a la educación, las herramientas como los sistemas de aprendizaje *on-line*, los asistentes inteligentes y la analítica del aprendizaje van a contribuir en la perso-

nalización del aprendizaje y extenderlo más allá de las paredes de un aula (Aika Educación, 2016). En este contexto, la IA ha tenido gran avance en el área de sistema de reconocimiento de voz, a tal grado de ser capaz de reconocer una solicitud y devolver una respuesta de manera natural como si fuera un humano (Esparza y Avalos, 2003).

Uno de los puntos a destacar en los sistemas de reconocimiento de voz es el aprendizaje automático de dichas inteligencias, ya que estas aprenden mediante patrones y reconocimientos que el usuario les da. Esto se debe a los algoritmos que toma del usuario, y pueden aprender de datos registrados y a realizar predicciones a partir de ellos (Amazon, 2018).

El concepto de reconocimiento de voz, mejor conocido como Reconocimiento Automático del Habla (RAH), es una parte importante de los sistemas de inteligencia artificial, cuyo objetivo principal es permitir la comunicación entre humanos y sistemas electrónicos (Postech IT Solution Provider, 2019). De acuerdo con Techopedia (2019), el sistema de reconocimiento de voz se define de la siguiente manera:

Es una técnica en tecnología informática mediante la cual se crean *software* y sistemas especializados para identificar, distinguir y autenticar la voz de un hablante individual. El reconocimiento de voz evalúa la biometría de voz de un individuo, como la frecuencia y el flujo de su voz y su acento natural. El reconocimiento de voz también se conoce como reconocimiento de altavoz (Techopedia, 2019).

Básicamente, los sistemas de reconocimiento de voz se usan para operar un dispositivo, ejecutar comandos o escribir sin tener que usar un teclado, *mouse* o presionar un botón. Hoy en día, esto se hace en una computadora o *smartphone* con programas de *software* como los de ASR, que en sus siglas en inglés significa “automatic speech recognition”. Muchos programas ASR requieren que el usuario “entrene” al programa para que reconozca su voz a fin de que pueda convertir el habla en texto con mayor precisión.

Los sistemas de reconocimiento están diseñados principalmente para reconocer la voz de la persona que habla. Las técnicas de reco-

nocimiento de voz requieren un entrenamiento en el cual el sistema subyacente aprenderá la voz, el acento y el tono del hablante, todo esto antes de reconocer la voz del hablante. Generalmente se logra a través de una serie de palabras y declaraciones textuales que la persona tiene que hablar a través del micrófono incorporado o externo (Techopedia, 2019).

Antes de que cualquier máquina pueda interpretar el habla, un micrófono debe traducir las vibraciones de la voz de una persona en una señal eléctrica en forma de onda. Esta señal, a su vez, la convierte el *hardware* del sistema, por ejemplo, la tarjeta de sonido de una computadora, en una señal digital; dicha señal digital la analiza un programa de reconocimiento de voz para reconocer fonemas separados, los componentes básicos del habla. Los fonemas se recombinan en palabras (Zwass, 2016).

Los sistemas de reconocimiento de voz están relacionados con los sistemas de reconocimiento del habla, pero el primero solo identifica al orador, mientras que el segundo puede entender y evaluar lo que se ha dicho. Es una tecnología que permite la entrada hablada en los sistemas. Al hablar con la computadora, teléfono o dispositivo, usa lo que dice el usuario como entrada para desencadenar una acción. Este tipo de tecnología se utiliza para reemplazar otros métodos de entrada como escribir, hacer clic o seleccionar de otras maneras, lo que lo convierte en un medio para hacer que los dispositivos y el *software* sean más fáciles de usar y aumentar la productividad (Unuth, 2019).

Hoy en día, con el creciente poder de las computadoras personales y dispositivos móviles, la precisión del reconocimiento de voz ha mejorado notablemente. Las tasas de error se han reducido a alrededor del 5 % en vocabularios que contienen decenas de miles de palabras (Zwass, 2016). Entre las primeras aplicaciones para el reconocimiento de voz se encontraban los sistemas telefónicos automatizados y el *software* de dictado médico. Con frecuencia se usa para dictados, para consultar bases de datos y para dar comandos a sistemas informáticos, especialmente en profesiones que se basan en voca-

bularios especializados. También permite asistentes personales en vehículos y teléfonos inteligentes, como el Apple Siri (Zwass, 2016).

Básicamente, existe una gran variedad de aplicaciones y áreas donde se utiliza el reconocimiento del habla, incluido el área militar, como una ayuda a las personas con discapacidad (imaginar a una persona sin manos o dedos), en el campo médico, en la robótica, telecomunicaciones, computación personal y otros (*Computer Hope*, 2019). Actualmente, casi todo el mundo está expuesto a estos sistemas debido a su propagación entre dispositivos comunes como computadoras y teléfonos móviles. Algunos teléfonos inteligentes como iPhone y Android están haciendo un interesante uso de los sistemas de reconocimiento de voz, a través de ellos pueden iniciar una llamada a un contacto con sólo recibir instrucciones por voz diciendo “Llamar a casa”. Otros comandos también pueden ser entretenidos, como “Encender la linterna” (*Computer Hope*, 2019).

En este contexto, uno de los dispositivos que utiliza sistemas de reconocimiento de voz son los altavoces inteligentes, los cuales se han ido popularizando sobre todo por la variedad de estos en el mercado por los diferentes fabricantes. De acuerdo con Silva (2019), se define un altavoz de la siguiente manera:

Es un dispositivo que no solo puede reproducir su música favorita, sino que puede proporcionar respuestas de preguntas introducidas verbalmente e incluso controlar partes de su hogar a través de una función de asistente virtual incorporada. Un altavoz inteligente amplía enormemente lo que normalmente pensamos como un sistema de reproducción de música.

Por lo anterior, la interfaz de lenguaje hablado a la computadora y ahora los dispositivos con altavoces (bocinas) que antes solo se usaban para reproducir música en altos volúmenes, las cuales poco a poco se han ido revolucionando agregando nuevas funcionalidades, haciéndolos dispositivos inteligentes, constituyen un tema que ha atraído y fascinado no solo a ingenieros científicos del lenguaje, sino también a todo usuario amante de la tecnología (Esparza y Avalos, 2003).

Por lo tanto, una bocina inteligente es un dispositivo inalámbrico (conectado mediante Wi-Fi) que reacciona ante órdenes emitidas oralmente y que está dotado de IA (Altavoz Inteligente, 2019). Se trata de un nuevo concepto de *gadgets* que además de poder emitir música, su vocación principal es la de escuchar y atender tus peticiones, que pueden ser sobre lo que sea (programar un recordatorio, una alarma, y otros). Su cerebro principal es un programa informático, un *asistente virtual*, que es capaz de entender la expresión del usuario y pronunciar las palabras naturalmente. Para esto, parte de una base de conocimientos que sumados a los complejos algoritmos basados en la tecnología de formación continua, lo llevan a aprender gustos, hábitos y aficiones, así como también del ecosistema familiar y doméstico (Sacristán, s/f).

En 2017 se realizó una encuesta sobre los usos comunes de un altavoz inteligente, realizada por Edison Research en EE. UU., la cual arrojó los datos siguientes: a) el 68 % de los encuestados lo utiliza para escuchar música, b) el 58 % para obtener información meteorológica, c) el 52 % para obtener respuestas de preguntas generales, d) el 45 % para escuchar noticias, e) el 43 % para programar alarmas y cronómetro, f) el 43 % para consultar la hora, g) el 33 % para controlar otros dispositivos inteligentes que tiene en casa, h) el 26 % para añadir a la lista de tareas, e i) el 26 % para añadir cosas a la lista de compra (Sacristán, s/f). En el mercado ya se encuentran diversos asistentes virtuales. En este contexto, Sacristán (s/f) publicó un artículo en la revista tecnológica *Gadget* sobre los “Altavoces Inteligentes: Qué son y para qué sirven”, donde mencionó:

Los principales asistentes virtuales hoy son Alexa (Amazon), Google Assistant, Siri (Apple), Cortana (Microsoft) y Bixby (Samsung). Menos el primero, ya conoces el resto por su presencia activa en los *smartphones* y ordenadores desde mucho antes de su integración en un altavoz (Alexa también se encuentra en cerraduras, automóviles, sistemas de iluminación, etcétera).

Por lo anterior, existen varios esfuerzos de asistentes virtuales que han desarrollado *skills* para ser implementados en diferentes contex-

tos. En el área educativa el asistente virtual de Amazon cuenta con múltiples aplicaciones en el entorno educativo: sirve como traductor en tiempo real, ayuda a hacer operaciones matemáticas, lleva un exhaustivo registro de las fechas de los exámenes (Moreno, 2018). A continuación, se muestran las *skills* de Alexa para aprender:

Lectura de audiolibros. En la actualidad, Alexa dispone de casi 50,000 *skills*; una parte de ellas adecuadas para este entorno. Los audiolibros ayudan a fomentar la imaginación y la concentración entre los estudiantes. Alexa tiene la capacidad para leer en voz alta todos los títulos comprados en la tienda de Amazon o los procedentes de los servicios Kindle Unlimited o Kindle Prime. Basta con registrarse en “Audible”, el servicio de audiolibros de Amazon, y lanzar con nuestra propia voz el comando para que Alexa lo reproduzca (Moreno, 2018).

Calculadora. Con tan solo decir en voz alta la operación matemática que se desea resolver, Amazon Alexa ofrece el resultado casi de forma simultánea. Está capacitado para trabajar con todo tipo de operaciones: desde sencillas sumas hasta operaciones como la raíz cuadrada o el seno y el coseno (Moreno, 2018).

Traductor en tiempo real. Traduce palabras y oraciones sencillas de manera simultánea, facilitando la conversación y comprensión en otros idiomas. Por ejemplo, cuando dices “Bienvenido a casa en inglés”, detecta de forma automática el idioma para traducir todo lo que se diga a partir de ese momento. También permite hacer conversión de divisas con tan solo preguntar: “¿Cuántos dólares son 50 euros?” (Moreno, 2018).

Lista de tareas. El asistente de Alexa lleva un registro exhaustivo, una agenda e, incluso, la actualiza, es suficiente con decirle la tarea que se quiere añadir y, de forma automática, aparecerá en el dispositivo móvil. Se podrán crear tantas tareas como se consideren necesarias: lista de deberes, fechas de los exámenes, los trabajos a entrega (Moreno, 2018).

Enciclopedia virtual. Alexa funciona también como diccionario y enciclopedia *on-line*. Así, rastrea toda la Web para ofrecer el mejor

significado de la palabra que se le pregunte a través de comandos (Moreno, 2018).

Material y métodos

El diseño de este proyecto de investigación es de tipo experimental, con un enfoque descriptivo y observacional. El tratamiento de la información se abordó con un enfoque cuantitativo y cualitativo. En cuanto al diseño y desarrollo de la *skill* del asistente inteligente de voz se tomó como referencia el ciclo de vida propuesto por Pressman (2010). Las fases de este modelo son cinco: a) análisis, b) diseño, c) desarrollo, d) prueba, y e) mantenimiento. A continuación, se describe lo realizado para el diseño de la *skill* del asistente inteligente de voz con base en el modelo de Pressman (2010).

Análisis. En esta fase se realizó un análisis Con base en los resultados del instrumento PPL con la finalidad de identificar los elementos y funciones que se integrarán a la *skill* del asistente inteligente de voz. Adicionalmente, se realizó un análisis de las características que tienen las bocinas inteligentes disponibles en la actualidad, para la selección de la bocina en la cual se diseñó dicha *skill*.

Diseño. En esta fase se diseñó la estructura de los atributos distintos de la interfaz y detalle procedimental (algoritmo). El proceso del diseño traduce requisitos en una representación del *software* donde se pueda evaluar su calidad antes de que comience la codificación.

Codificación. En esta fase el diseño se tradujo en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Esta fase se consideró de las más importantes ya que gracias a esta se llevarán a cabo cada una de las tareas para las que está diseñada la *skill* del asistente inteligente de voz.

Prueba. En esta fase se generó el código, y se comenzó con las pruebas de la *skill* del asistente de voz, la cual se centró en los procesos lógicos internos del *software* asegurando que todas las sentencias se han comprobado y sean funcionales. Es decir, en esta

fase se realizaron las pruebas para la detección de errores, se aseguró que la entrada definida produjera los resultados reales de acuerdo con los resultados requeridos.

Mantenimiento. En esta última fase, la *skill* del asistente inteligente de voz sufre cambios porque el *software* debe adaptarse para acoplarse a los cambios de su entorno externo; en dicha fase, Con base en su implementación y con la aplicación de un instrumento de evaluación de la *skill* se detectaron mejoras funcionales o de rendimiento. El soporte y mantenimiento del *software* vuelve a aplicarse a cada una de las fases precedentes.

Resultados

En este apartado, se presentan los resultados Con base en las cinco fases del modelo propuesto por Pressman (2010).

Fase 1. Análisis

Con base en los resultados obtenidos de la aplicación del cuestionario PPL se consideró integrar a la *skill* del asistente de voz los siguientes elementos: a) configuración de comandos más comunes, b) uso de lenguaje común, c) integración de deletreo de comandos, y d) idioma español. En lo que respecta a la selección de la bocina se eligió Alexa de Amazon, ya que ofrece un conjunto de herramientas API (cuyas siglas significan “*application programming interface*”) con soluciones de referencia y documentación, por lo que las creaciones de las *skills* se pueden diseñar de forma muy sencilla. Asimismo, se eligió este dispositivo en específico por su accesibilidad en código abierto, además de las siguientes características en cuanto a *hardware* y *software*: a) su diseño se adapta a todas las situaciones por su reducido peso y dimensiones, b) calidad de sonido, c) es más económico, d) tiene variedad de comandos interesantes y aprende constantemente

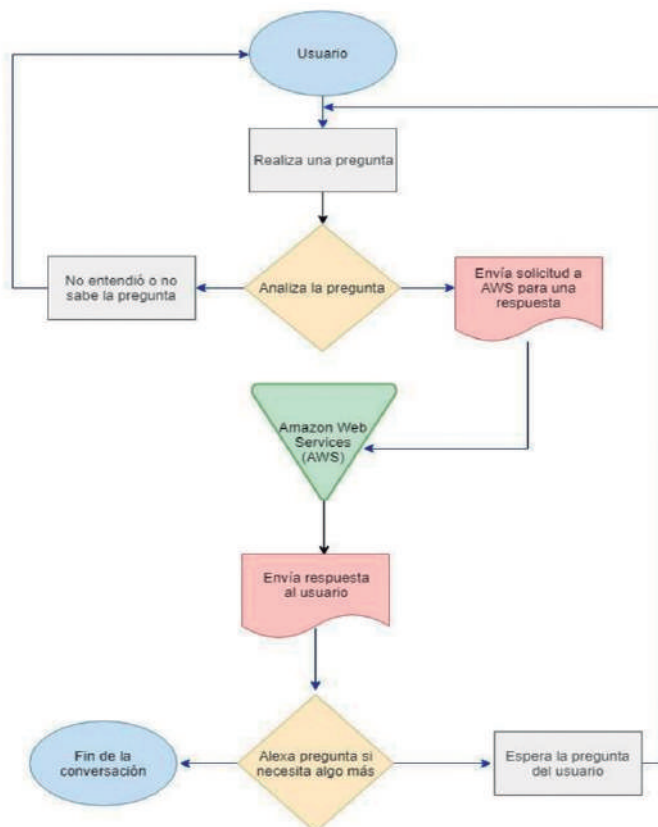
nuevos, e) tiene un asistente de voz completo, f) gran resistencia, g) no requiere programación compleja, y h) plataformas sencillas y fáciles de entender.

Fase 2. Diseño

En esta etapa se diseñaron los diagramas de flujo de la *skill* del asistente de voz. La figura 6 muestra la manera en que funciona el asistente.

Figura 6

Diagrama de flujo. Funcionamiento de la *skill* del asistente inteligente



En las figuras 7 y 8 se muestran los diagramas de flujo del proceso cuando un usuario solicita una pregunta respecto a los comandos. Se ejemplifican dos comandos en los diagramas de flujo, primero sobre asignar una dirección IP, y el segundo sobre configurar un nombre al dispositivo de red.

Figura 7

Diagrama de flujo. Proceso de respuesta de Alexa 1

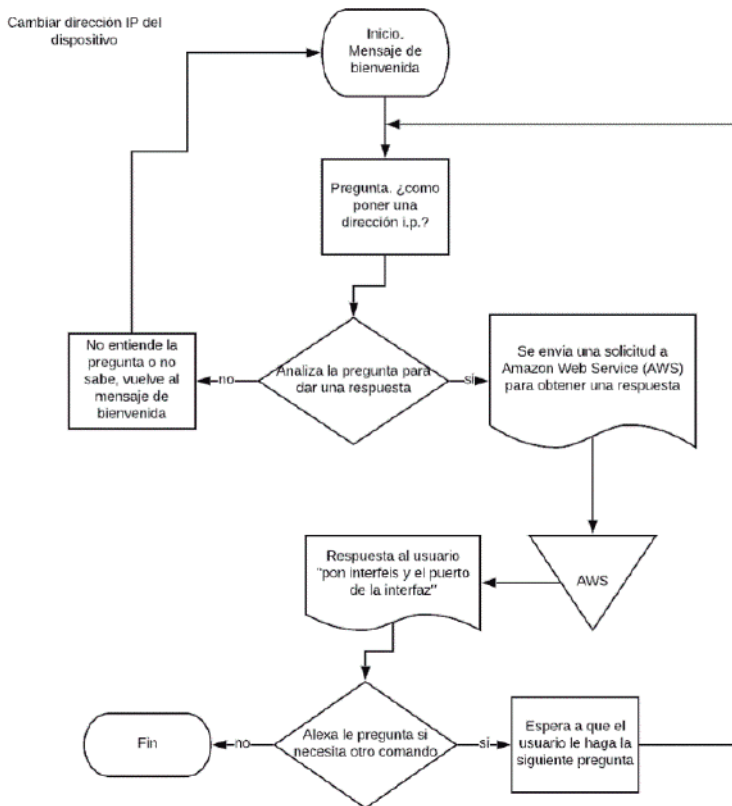
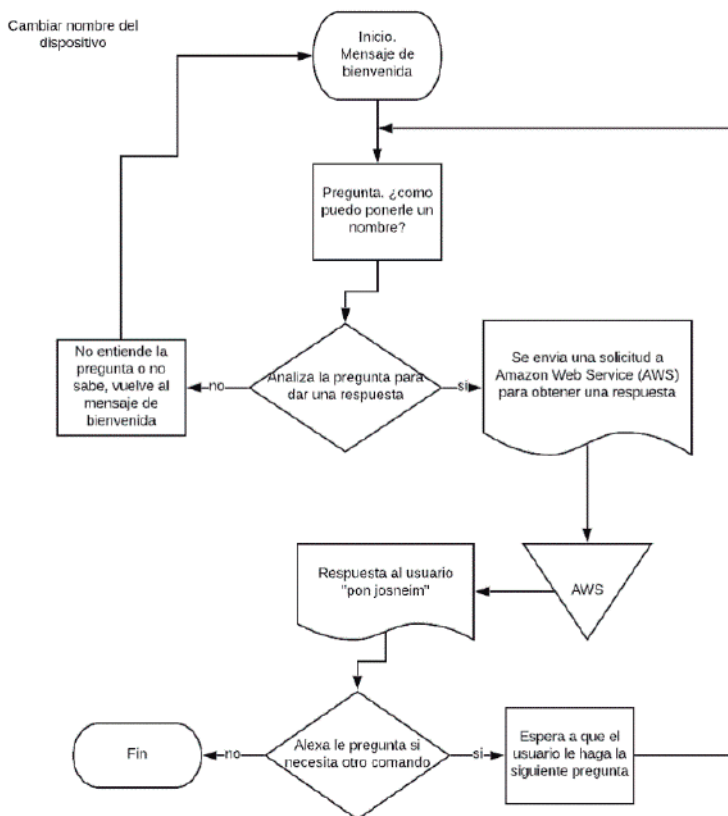


Figura 8

Diagrama de flujo. Proceso de respuesta de Alexa 2



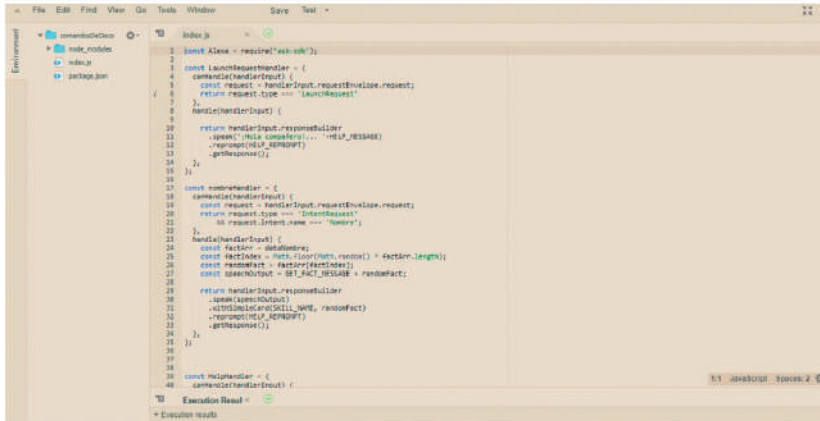
Fase 3. Codificación

Para la elaboración de este asistente de voz, se tienen dos plataformas para su diseño y construcción. Una de ellas es la plataforma Amazon Developer Console, en la cual se pueden crear los bancos de preguntas y respuestas posibles que el usuario pueda dar. Cabe a destacar que todas las funciones que se realicen en este sitio tienen que ser programadas en otra plataforma que Amazon ofrece, cuyo nombre

es Lambda. En la figura 9 se puede observar el código realizado en la plataforma de Lambda.

Figura 9

Código realizado en la plataforma Lambda Management Console



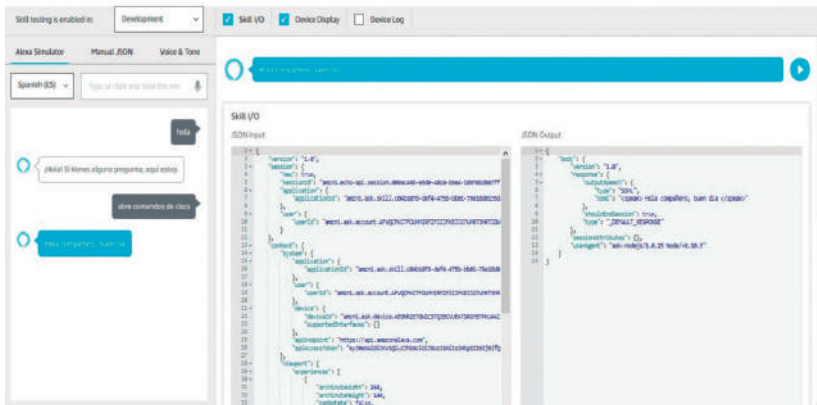
```
1 const Alexa = require('ask-sdk');
2
3
4 const LaunchRequestHandler = {
5   canHandle(handlerInput) {
6     const request = handlerInput.requestEnvelope.request;
7     return request.type === 'LaunchRequest';
8   },
9   handle(handlerInput) {
10
11
12     return handlerInput.responseBuilder
13       .speak('¡Hola compañero!... ' + HELP_MESSAGE)
14       .getResponse();
15   }
16 };
17
18 const HomeRequestHandler = {
19   canHandle(handlerInput) {
20     const request = handlerInput.requestEnvelope.request;
21     return request.type === 'IntentRequest';
22   },
23   handle(handlerInput) {
24     const factId = getFactId();
25     const factIndex = Math.floor(Math.random() * factId.length);
26     const randomFact = facts[factIndex];
27     const utterance = GET_FACT_MESSAGE + randomFact;
28
29     return handlerInput.responseBuilder
30       .speak(utterance)
31       .withIntent(handlerInput.getIntent().name, randomFact)
32       .getResponse();
33   }
34 };
35
36
37 const HelpRequestHandler = {
38   canHandle(handlerInput) {
```

Fase 4. Pruebas

Antes de instalar la *skill* a la bocina Alexa Echo, se comprobó y realizó una prueba, para ver los resultados y el funcionamiento correcto de la inteligencia. En la figura 10 se muestra la prueba realizada.

Figura 10

Prueba realizada en la plataforma de Alexa Developer Console



En esta prueba, cuando el usuario interactúa con la *skill* inteligente de voz, lo que hace es mandar una solicitud a la API en los servicios de Lambda, en la cual se ha programado, y, en dicho código, genera la respuesta o la interacción que la bocina Alexa va generar para que el usuario pueda preguntar y consultar los comandos que necesite. Esto se muestra en la figura 11.

Figura 11

Entrada y salida de la solicitud para una respuesta al usuario



Adicionalmente, las pruebas realizadas, desde la plataforma de Amazon Developer Console, se pueden observar en la figura 12.

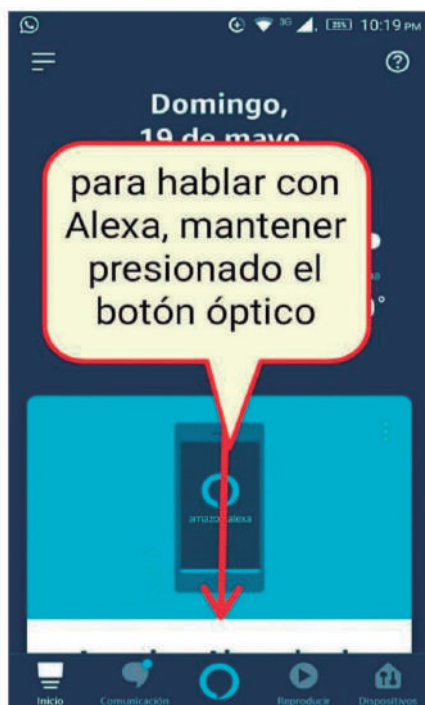
Figura 12
Resultados de las pruebas



Fase 5. Implementación

En esta fase se realizó la implementación final utilizando la bocina de Alexa. Asimismo, cabe mencionar que el usuario puede descargar una aplicación para interactuar con la *skill* del asistente inteligente de voz en un dispositivo móvil o en su computadora, sin la necesidad de tener que tener en físico la bocina Alexa. La figura 13 muestra una imagen de las pruebas reales realizadas desde la aplicación de Alexa Amazon.

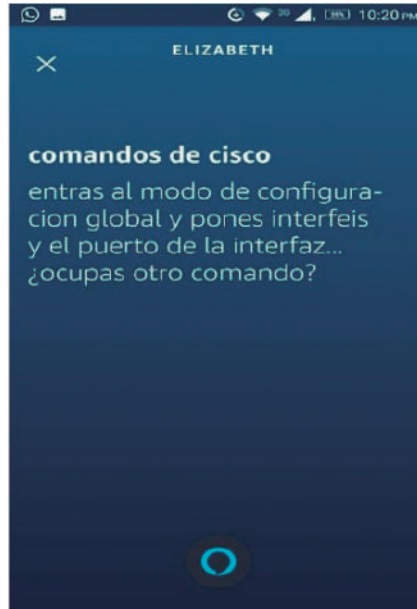
Figura 13
Interfaz principal de la aplicación Alexa Amazon



Al momento de preguntarle algo a Alexa, dará una respuesta a la pregunta realizada, como se puede ver en la figura 14.

Figura 14

Ejemplo de una respuesta que Alexa provee



Si el usuario se dirige al apartado “Skills y Juegos”, se pueden encontrar todas las *skills* desarrolladas, o bien, descargadas e instaladas, en las cuales se pueden hacer pruebas en tiempo real desde el móvil, como se observó en las ilustraciones anteriores.

Discusión y conclusiones

Con el desarrollo de la *skill* del asistente inteligente de voz como apoyo en la educación, se confirmó que es un material didáctico innovador para los alumnos de la ITEL del Centro Universitario de la Costa de la Universidad de Guadalajara, México. Adicionalmente, con la implementación de este material didáctico se pudo concluir lo siguiente: Primero, este material didáctico ayudó a los alumnos de la

ITEL para fortalecer su proceso de enseñanza-aprendizaje. Segundo, con el diseño de esta *skill* se pudo demostrar que puede funcionar para el área educativa. Tercero, con este material educativo se integró una tecnología actual que cada vez más gana presencia en el mercado mundial. Finalmente, se concluyó que de este material educativo fue pertinente su creación para lograr una de las metas establecidas en el *Plan Desarrollo Institucional de la Universidad de Guadalajara 2014-2030*, donde se establece impulsar el uso de las tecnologías de la información para fortalecer el proceso de enseñanza de los alumnos.

Con base en lo anterior, se puede dar continuidad a proyectos futuros, como desarrollar la *skill* del asistente inteligente de voz en su versión en inglés para promover la práctica del segundo idioma.

Adicionalmente, este proyecto puede crecer para incluir más comandos de configuración de los equipos de Cisco, ya que el asistente inteligente de voz tiene la limitante de incluir los comandos más usados. Asimismo, se pueden desarrollar otros proyectos similares integrando la metodología de su desarrollo para el diseño de otros materiales didácticos. Finalmente, se puede trabajar en un proyecto diseñado a manera de juego, para hacerlo más interesante y divertido.

Referencias

- Aika Educación (2016), “Inteligencia artificial: las tecnologías que cambiarán la educación en 2030”, 30 de noviembre. Disponible en: <http://www.aikaeducacion.com/tendencias/inteligencia-artificial-las-tecnologias-cambiaran-la-educacion-2030/>
- Altavoz Inteligente (2019), “Altavoz inteligente”. Disponible en: <https://altavozinteligente.net/>
- Amazon (2000), “¿Qué es la inteligencia artificial?”, Amazon Web Services. Disponible en: https://aws.amazon.com/es/machine-learning/what-is-ai/?fbclid=IwAR0EWGaterDo8OopRTGzJNyNse_UEgnyw5FQXKLXvhAYGiYtcnspArWdsA
- (2018), “Amazon Alexa”, Developer Amazon. Disponible en: <https://developer.amazon.com/es/alexa>

- Centro Universitario de la Costa (2019a), “Historia”, CUCosta UdeG. Disponible en: <http://www.cuc.udg.mx/historia>
- (2019b), “Misión Telemática”, CUCosta. Disponible en: <http://www.cuc.udg.mx/es/mision-15>
- (2019c), “Oferta académica”. Disponible en: <http://www.cuc.udg.mx/es/oferta-academica-cucosta>
- Cisco Networking Academy (2020), “Cisco Networking Academy”. Disponible en: <https://www.netacad.com/>
- Computer Hope (2019), “Digital Assistant”, 31 de enero. Disponible en: <https://www.computerhope.com/jargon/d/digital-assistant.htm>
- Cruzart, S. (2009), “Materiales didácticos en organizadores visuales”, diciembre. Disponible en: <http://es.scribd.com/doc/17288813/MATERIALES-DIDACTICOS>
- “Dictamen de Creación del Plan de Estudios de la Licenciatura en Ingeniería en Telemática” (2017), Universidad de Guadalajara.
- Esparza, M. E. y J. B. Avalos (2003), “Reconocimiento de voz”, *Conciencia Tecnológica*, (22). Disponible en: <http://www.redalyc.org/articulo.oa?id=94402206>. Consultado: julio, 2019.
- López, R. O. (2006), *Medios y Materiales Educativos*, Perú, Universidad Nacional Pedro Ruiz Gallo, Facultad de Ciencias Histórico Sociales y Educación.
- Moreno, M. (2018), “Saca el máximo partido a Alexa con estos usos educativos”, *Educación 3.0*, 4 de diciembre. Disponible en: <https://www.educacionrespuntocero.com/recursos/usos-educativos-alexa/95602.html>.
- Postech IT Solution Provider (2019), “Postech”. Disponible en: <http://postech.com.mx/Postech/ES/voz.php>
- Pressman, R. S. (2010), “Ingeniería de *software*: Un enfoque práctico”, en R. S. Pressman, *Parte 1. El proceso del software: Capítulo 2. Modelos del proceso*, pp. 33-35, México, DF, McGraw-Hill.
- Ramírez P., M. (2011), “Informática para la educación”, *La educación al alcance de tu móvil: Mobile Learning (M-learning)*, 25 de octubre. Disponible en: <https://milagrosrp.wordpress.com/tag/m-learning/>
- Silva, R. (2019), “¿Qué es un altavoz inteligente?”. Disponible en: <https://www.lifewire.com/smart-speaker-4145037>
- Simonit, J. (2009), “Materiales educativos”, 24 de febrero. Disponible en: <http://coleccion.educ.ar>

- Techopedia (2019), “Voice Recognition”. Disponible en: <https://www.techopedia.com/definition/9961/voice-recognition>
- Universidad de Guadalajara (2010), *Plan de Desarrollo Institucional 2030*. Disponible en: <http://www.copladi.udg.mx/sites/default/files>
- (2019), “Historia”, CUCosta, UdeG. Disponible en: <http://www.cuc.udg.mx/historia>
- Unuth, N. (2019), “What Is Speech Recognition?”, *Lifewire*, 23 de abril. Disponible en: <https://www.lifewire.com/what-is-speech-recognition-3426721>
- Valois, M. A. (2018), “Alexa y Siri: ¿Sabes qué es un asistente virtual inteligente?”, *Hostgator*, 20 de septiembre.
- Zampayo, C. F. (2004), *Técnicas y métodos para la representación y manipulación del conocimiento en ciencias computacionales (Tesis)*, México, UNAM.
- Zwass, V. (2016), “Speech Recognition”, *Britannica*, 10 de febrero. Disponible en: <https://www.britannica.com/technology/speech-recognition>

2

SISTEMA DE INFORMACIÓN WEB PARA EL REGISTRO Y SEGUIMIENTO A EMPREENDEDORES DE UNA INCUBADORA DE NEGOCIOS*

JULIO FRANCISCO BAHENA AYALA¹

AURELIO ENRIQUE LÓPEZ BARRÓN²

MARÍA DEL CONSUELO CORTÉS VELÁZQUEZ³

KARINA DONLUCAS MAGAÑA⁴

Resumen

Actualmente, en México se encuentran registradas 197 incubadoras de negocios ante el Instituto Nacional del Emprendedor (INADDEM). En el estado de Jalisco se encuentran reconocidas cinco incubadoras básicas, tres de ellas en Guadalajara, una en Tequila y otra

* Derivado del proyecto de investigación: "Actividades de Investigación del Cuerpo Académico UDG-CA-884".

1 Universidad de Guadalajara, correo electrónico: m_jay3@hotmail.com

2 Universidad de Guadalajara, correo electrónico: aurelio.lbarron@academicos.udg.mx

3 Universidad de Guadalajara, correo electrónico: maria.cortes@academicos.udg.mx

4 Universidad de Guadalajara, correo electrónico: karina.donlucas@academicos.udg.mx

en Puerto Vallarta reconocida como “Vallarta Emprende AC”, la cual es objeto de estudio de este proyecto de investigación (INADEM, 2015). La incubadora de negocios Vallarta Emprende se constituyó en el año 2012 y una de sus tareas principales es llevar a cabo el registro de los emprendedores quienes tienen el propósito de mejorar su empresa. Dicho registro es realizado en hojas de cálculo, lo que implica que estas actividades se realicen de manera no eficiente al hacer el proceso muy lento, poco accesible y con una nula seguridad. Por lo anterior expuesto, el objetivo de este proyecto de investigación es diseñar y desarrollar un sistema web para el registro y seguimiento de los empleadores de la incubadora de negocios Vallarta Emprende. En este sentido, dicho sistema se denominó “Sistema Emprendedor” y fue creado tomando como base la metodología de ingeniería basada en la Web (UWE)-UML propuesta por la Dra. Nora Koch (2010), la cual incluye seis etapas: a) análisis de los requisitos, b) diseño, c) codificación, e) pruebas, f) implementación, y g) mantenimiento.

Palabras clave: sistema de información, sistema web, modelos de desarrollo web.

Abstract

Currently, in Mexico 197 business incubators are registered with the National Entrepreneur Institute (INADEM). In the state of Jalisco, five basic incubators are recognized, three of them in Guadalajara, one in Tequila and another in Puerto Vallarta recognized as “Vallarta Emprende AC” which is the object of study of this research project (INADEM, 2015). The Vallarta Emprende business incubator was established in 2012 and one of its main tasks is to carry out the registration of entrepreneurs who have the purpose of improving their company. Said registration is carried out in spreadsheets, which implies that these activities are carried out in an inefficient way by making the process very slow, not very accessible and with no security. Therefore, stated the objective of this research project

is to design and develop a web system for the registration and monitoring of the employers of the Vallarta Emprende Business Incubator. In this sense, this system was called “Entrepreneurial System” and it was created based on the Web-based engineering (UWE)-UML methodology proposed by Dr. Nora Koch (2010) which includes six stages: a) requirements analysis, b) design, c) coding, e) testing, f) implementation, and g) maintenance.

Introducción

Una incubadora de negocios es un centro de atención a emprendedores que tiene la finalidad de orientar y asesorar para llevar a cabo un negocio real. Además, dichas incubadoras brindan el apoyo para elaborar un plan de negocio para la creación de una empresa, proporcionándote consultoría en las diversas áreas que necesitas manejar al ser empresario: mercadotecnia, contabilidad, diseño gráfico e industrial (INADEM, 2015). En México, actualmente existen 235 incubadoras reconocidas por el Instituto Nacional del Emprendedor, de las cuales 180 son incubadoras básicas, 28 incubadoras de alto impacto, dos espacios de vinculación y 25 aceleradoras. Todas estas incubadoras pertenecen a la red de apoyo al emprendedor Para Mover a México que asesora a aquellos emprendedores y empresarios que desean iniciar su propio negocio (INADEM, 2015). En el estado de Jalisco se encuentran reconocidas trece incubadoras, las cuales son las impulsoras de la generación de empresas y empresarios exitosos en nuestro estado, la mayoría se encuentran en la capital del estado y en la ciudad de Puerto Vallarta se localizan dos de estas: Vallarta Emprende AC y el Centro de Desarrollo Empresarial UNIVA Vallarta. Algunas incubadoras en México ofrecen sus propias oficinas para que el emprendedor practique en ellas la implementación de su propia empresa y entrene durante su proceso de incubación para enfrentarse al mercado y competencia cuando arranque su negocio. Igualmente, durante este proceso los consultores de la incubadora les

brindan a los emprendedores la capacitación, asesoría y consultorías especializadas necesarias para que este aprenda todos los requisitos y retos que necesita superar para llevar a su empresa al éxito (INADEM, 2015). La incubadora de negocios Vallarta Emprende AC, donde se desarrolló el proyecto de esta investigación, se constituyó formalmente el 31 de mayo del año 2012 en Puerto Vallarta, Jalisco. El objetivo de esta incubadora de negocios es

fomentar la creación de nuevas micro y pequeñas empresas, con el fin de generar más y mejores empleos e inculcar la cultura de emprendimiento en México, pero dirigido a la población de nuestra región Costa Norte y Costa Sur, trabajando para elevar la competitividad.

Sus objetivos a corto plazo son: a) brindar servicios de consultoría a por lo menos 20 empresas, b) lograr al menos 20 % de empresas egresadas financiadas por la vinculación a financiamiento, y c) Certificación de la National Business Incubation (NBIA) para la incubadora y consultores. De igual manera, los objetivos a mediano plazo son: a) la atención de emprendedores y empresas en la región Costa Sur, b) lograr financiamiento para el 60 % de las empresas egresadas, y c) aumentar el número de proyectos atendidos en seguimiento en al menos un 50 %. Los objetivos a largo plazo son: a) la atención de emprendedores y empresas en la región Sierra Occidental, b) lograr la supervivencia de por lo menos el 95 % de las empresas incubadas, y c) lograr la implementación de tecnologías para la mejora de servicios en las MiPyME (Vallarta Emprende, 2015). Asimismo, la misión de la incubadora de negocios de Vallarta es “formar emprendedores y empresarios, con vocación de servicio cuyo desarrollo sea integral, genere empleos dignos y promueva la competitividad en la Costa del estado”. En cuanto a su visión, es “ser una incubadora básica con un alto porcentaje de empresas exitosas en cuanto a rentabilidad, durabilidad y generación de empleos de calidad” (Vallarta Emprende, 2015).

Actualmente, la incubadora de negocios Vallarta Emprende AC apoya a emprendedores y empresarios a iniciar e impulsar su propio negocio, por lo que se llevan a cabo registros de los datos princi-

pales de cada uno de ellos, también de aquellos empresarios que quieren mejorar su empresa. La información básica y relevante de esta incubadora de negocios se almacena en hojas de cálculo, lo que implica que la información de una sola persona se repita en varias ocasiones y al momento de hacer reportes no salga con la exactitud correcta. Lo anterior repercute en un problema al momento de contabilizar el total de personas emprendedoras que se tienen en determinado momento y con las cuales se trabajará, ya que la información no se encuentra centralizada y es muy susceptible a errores como la omisión de información (González, comunicación personal, 1 de mayo de 2015). Adicionalmente, no se lleva un método exacto para registrar los documentos que los emprendedores entregan, por lo que en ocasiones se llega a repetir información que antes ya se había registrado, o en algunos casos no entregan la documentación correspondiente. Como consecuencia de lo anterior, se ha tenido la problemática de que varios emprendedores queden fuera de las convocatorias al no entregar su información a tiempo o completa, lo que conlleva a que tengan que esperar la siguiente convocatoria para poder participar de nuevo. Finalmente, los encargados de capturar la información de los emprendedores en la incubadora de negocios de Vallarta se han visto en la necesidad de tener que trabajar horas extras para poder organizar la información recibida de los emprendedores, desaprovechando tiempos valiosos que pueden ser usados en otras actividades que pueden resultar más productivas, por ejemplo, se podría apoyar a más personas que desean poner en marcha su empresa, obtener mejores resultados para la aprobación de los negocios y más historial favorable para estos organismos (González, comunicación personal, 1 de mayo de 2015).

Por lo anterior, el propósito de este proyecto de investigación fue diseñar y desarrollar un sistema de información web para el registro y seguimiento a emprendedores de una incubadora de negocios: Vallarta Emprende, denominado “Sistema Emprendedor”. Lo anterior se justifica porque contar con un sistema de información (en adelante SI) permitirá a la empresa tener mayor control, seguridad,

administración y eficiencia en sus procesos. Finalmente, poseer un SI permite tener las siguientes ventajas: a) ahorro de dinero, al no tener que comprar un *software*; b) uso flexible, al no tener que contemplar costos de servidores; c) no se requiere pagar por un soporte de forma definitiva; y d) se puede tener acceso a todo momento, desde cualquier lugar (Luke, 2015). De acuerdo con lo anterior, se diseñaron las siguientes preguntas de investigación:

1. ¿Cuáles son los requisitos para el Sistema Emprendedor?
2. ¿Cuál es la codificación del Sistema Emprendedor?
3. ¿Qué pruebas aplicarán en el Sistema Emprendedor?
4. ¿Cuál será el proceso de implementación del Sistema Emprendedor?
5. ¿Cómo será el proceso de mantenimiento del Sistema Emprendedor?

Revisión de la literatura

Para comprender el concepto de SI, es necesario conocer los conceptos básicos que conllevan a dicho sistema. Adicionalmente, identificar los componentes que ayudan a relacionarse con él y las diferentes clasificaciones populares que existen. A continuación se presentan algunas conceptualizaciones de sistemas y posteriormente la conceptualización de SI.

Un sistema es un grupo de elementos que interaccionan entre sí, para lograr un objetivo determinado. Aunque existe una gran variedad de sistemas, la mayoría de ellos pueden representarse a través de un modelo formado por cinco bloques básicos: elementos de entrada, elementos de salida, sección de transformación, mecanismos de control y objetivos (Fernández, 2006, p. 11).

Por otro lado, un sistema se define como un “conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí” (Real Academia Española, 2012).

De manera específica, los SI “son un sistema que, actuando en un conjunto de datos, los recopila, procesa y suministra como información allí donde sea necesaria para la actividad empresarial...”, tal y como lo mencionó Olson (1999, citado en Salvador, 2003). De manera similar, Peña (2006) en su tesis doctoral señaló que un SI es un conjunto de elementos interrelacionados con la finalidad de poner atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. Por lo anterior, se puede concluir que los SI son la base e importancia de una organización, ya que con ellos se puede agilizar de una manera notable el rendimiento del trabajo efectuado, además, ayudan a reducir el tiempo de atención a un cliente.

Los elementos que conforman un SI de acuerdo con Pérez y Mendoza (2014) son cuatro: a) la información, b) personas, c) tecnología de información y comunicación, y d) las técnicas de trabajo. De acuerdo con los anteriores autores, la información es la más importante del tema, ya que con ella se desarrollan las diversas maneras de administrarla y consultarla. La información es un conjunto de hechos organizados de tal manera que poseen un valor adicional más allá del valor que se les puede atribuir como hechos individuales. Las personas son los encargados de introducir, procesar y/o utilizar la información almacenada del sistema. Este elemento está compuesto por dos grupos: el técnico y el usuario. El técnico, que posee los conocimientos especializados en el desarrollo de sistemas, siendo estos los administradores, líderes de proyecto, analistas, programadores, operadores y capturistas. El usuario, representado por las personas interesadas en el manejo de información vía cómputo, como apoyo al mejor desempeño de sus actividades, siendo estos los funcionarios, contadores, ingenieros, empleados, público y otros (Stair y Reynolds, 2010). En cuanto al cuarto elemento, Peña (2006) mencionó que la tecnología de información y comunicación es un elemento que se integra por un listado de herramientas de *hardware* y *software*, los cuales son desarrollados para administrar la información

y enviarla de un punto a otro. Finalmente, las técnicas de trabajo se pueden definir como los procesos utilizados por las personas y las tecnologías para el desarrollo de sus actividades.

A diferencia de lo anterior expresado, los autores Stair y Reynolds (2010) mencionaron que los componentes de un sistema son tres: a) entrada, b) procesamiento, y c) salida. De acuerdo con los elementos mencionados por estos autores, la entrada se define como la actividad consistente en la recopilación y captura de datos. El procesamiento significa la conversión o transformación de datos en salida útiles. La conversión de datos en información útil es un aspecto crítico en escenarios de negocios. El procesamiento puede hacerse de forma manual o por medio de una computadora. La salida involucra la producción de información útil, por lo general en la forma de documentos y reportes. También los autores antes mencionados refirieron que en un sistema se puede contar con un módulo de retroalimentación, que es la información proveniente del sistema que se utiliza para realizar cambios en las actividades de entrada y de procesamiento. La retroalimentación es parte importante para los administradores y las personas que toman las decisiones en la organización.

Con base en lo anterior, se puede decir que los SI son indispensables para los organismos, ya que ayudan a centralizar la información. Además, en los últimos tiempos se ha visto un incremento en la necesidad de tener la información de una manera más rápida, fluida y accesible. En este contexto, Fleitman (2000) mencionó que en la actualidad ya no funcionan las formas antiguas de hacer negocios. Los procesos de administración, producción y distribución de artículos, así como los servicios, se han vuelto más dinámicos y complicados. También remarcó que la tecnología es indispensable con el modo de hacer negocios, ya que sin la información precisa las empresas corren el riesgo de no tomar las decisiones correctas. La información ha pasado a ser la clave de la toma de decisiones de una empresa, ya que con ella se pueden tomar acciones a futuro o estrategias para mejorar la producción y calidad de su producto o servicio. Finalmente,

este autor remarcó que los sistemas son clave para los negocios y que ayudan a que la información sea sencilla, transparente, libre, verídica, exacta, congruente y fácil de analizar e interpretar.

Según Laudon y Laudon (2016), existen diferentes tendencias, propiedades y niveles en una organización, por lo que se pueden contar con diferentes tipos de sistemas. En este sentido, los SI se clasifican en los siguientes: a) sistemas a nivel operativo, b) sistemas a nivel del conocimiento, c) sistemas a nivel administrativo, d) sistema a nivel estratégico, e) sistema para procesamiento de transacciones, f) sistemas de soporte para la decisión, g) sistemas de información para la toma de decisiones en grupo, h) sistemas de trabajo con conocimientos, e i) sistemas de automatización de oficinas.

Como se abordó en temas anteriores, los SI son la base de muchos de los logros que puedan llegar a tener las organizaciones ya que interactúan directamente con la toma de decisiones. Pero la demanda de estos sistemas ha proliferado de tal manera que ya son necesarios en cada negocio, y el acceso rápido a la información es indispensable en todo momento. Para esto, es necesario el uso de Internet. A continuación se abordan las siguientes conceptualizaciones que se relacionan e intervienen de forma directa con un sistema de información con base web: a) conceptualización de Internet, b) conceptualización de Web, y c) conceptualización de protocolos. De acuerdo con Resino y Ena (2007), Internet es una red mundial de computadoras interconectadas entre sí. Esta interconexión permite intercambiar datos entre ellas, además de compartir recursos. De igual manera, González (2010) mencionó que Internet es una red internacional creada en 1969 como un experimento militar norteamericano para dar seguridad a sus comunicaciones. La www, comúnmente conocida como Web, actualmente es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo. La Web es el área de Internet que se ha desarrollado con mayor auge. Con base en ello, una gran cantidad de hackers y desarrolladores de todo el planeta, empezaron a desarrollar sus propios navegadores (*browsers*) apoyados con el trabajo de Berners-Lee. De igual manera, Canelo

y Alonso (2007) añadieron que un navegador web es un programa informático que facilita la interacción con un servidor para acceder a los recursos de Internet e interpretar las etiquetas de los documentos a los que se hace mención anteriormente. Finalmente, de acuerdo con Luján (2002), se crearon diversos protocolos para Internet, de los cuales los más comunes son: HTTP, HTTPS, DNS, FTP, entre otros.

De acuerdo con Granados (2014), una aplicación web requiere de una base de datos para poder operar. Hay diferentes tecnologías, pero usualmente MySQL es la mejor opción para la mayoría de los casos. MySQL es un sistema de administración de bases de datos relacionales, es rápido, sólido y flexible. Es la manera ideal de crear bases de datos con acceso desde una página web, para el desarrollo de sistemas de transacciones *on-line* o para cualquier otra implementación que implique almacenar datos (Cobo, Gómez y Rocha, 2005). Por lo anterior, una base de datos es un grupo de datos almacenados sin repeticiones innecesarias en un soporte informático y con acceso a ella, al mismo tiempo, por diferentes usuarios y aplicaciones. Los datos deben de estar organizados y recopilados de una forma independiente de las aplicaciones que acceden a ella (Cobo, 2007). De igual manera, los autores Pons, Marín, Medina, Acid y Vila (2009) mencionaron que una base de datos es un núcleo general de información reunida en un ordenador para que cualquier usuario o programa con la autorización correcta pueda acceder a ella, indistintamente de la procedencia que tenga y el uso que le dé. En este contexto, hay diferentes tipos de bases de datos, y se clasifican según la utilidad que se les estará dando. Para esto hay un método llamado modelo de datos, que nos ayuda a representar los datos que se van a almacenar en forma de símbolos, conceptos y normas a seguir. Aunque pueden variar, los siguientes elementos se pueden encontrar en la mayoría de los modelos de datos: a) entidades, b) atributos o propiedades de las entidades, c) dominios o conjuntos de valores sobre los que se describen los atributos, y d) relaciones o asociaciones entre objetos (Piñeiro, 2013). Según la variabilidad de los datos, se pueden clasificar en bases de datos estáticas y bases de datos dinámicas. En las

bases de datos dinámicas la información que es almacenada se puede modificar, posibilitando acciones como actualización, eliminación e introducción de datos, además de las consultas. Las bases de datos estáticas son de solo lectura, empleada principalmente para almacenar datos históricos que en su momento pueden ser utilizados para analizar los datos al paso del tiempo o para hacer toma de decisiones sobre una información en específico (López, 2011). Los modelos de las bases de datos se pueden clasificar en bases de datos jerárquicas, relacionales, de red y transaccionales; aunque existen muchas otras, son las más utilizadas (Piñeiro, 2014).

La implementación de la ingeniería del *software* facilita conseguir programas de calidad y eficaces, permitiendo que los resultados se ajusten a las necesidades existentes del usuario. Esta particularidad de la ingeniería del *software* ha permitido que los SI pudieran evolucionar a los nuevos entornos de hipermedia y web. Esto fue el auge de lo que en la actualidad se conoce como ingeniería web (Deshpande, Murugesan, Ginige, Hansen, Schwabe Gaedke y White, 2002). A continuación se mencionan las metodologías usadas para un SI web. Es importante mencionar que los modelos para la metodología en un SI son muy recientes en su área, la mayoría de las ofertas planteadas a continuación están en constante revisión y podrían cambiar para favorecer su estructura.

Hypermedia Design Model o *Modelo de Diseño de Hipermedia (HDM)*. De acuerdo con Chen (1976), fue la primera opción que se desarrolló con el fin de dar apoyo al proceso de SI hipermedia. No es una metodología en sí, en realidad el HDM es una prolongación del modelo entidad-relación en el cual se anexan nuevas características para el modelado de sistemas hipermedia. El HDM propone un grupo de componentes que ceden al diseñador detallar una aplicación. Estos elementos son las entidades, los componentes, las perspectivas, las unidades y los enlaces. Los elementos antes mencionados pueden encontrarse en el estudio del clásico entidad-relación. Así pues, una aplicación basada en el modelo HDM se apoya en un arreglo general compuesto por unidades básicas llamadas entidades. Una entidad

indica un objeto físico o conceptual de la aplicación. En el HDM las entidades son reunidas en tipos de entidad. Los tipos de entidad se identifican por un nombre, por un grupo de perspectivas en las cuales se puede observar su contenido y un conjunto de enlaces de aplicación por los que se puede navegar. Una entidad es la unidad más pequeña y autónoma de cualquier modelo HDM. Cada entidad está formada por una categoría de componentes que heredan las características de dicha entidad. Los componentes no sirven de nada si no existe una entidad a la que dependa. Los componentes son contemplaciones para diseñar un conjunto de unidades o nodos que constituyen un mismo grupo de información de una entidad. Una unidad es un almacén de información contenida de la aplicación. Asimismo, una unidad incorpora un segmento de contenido de una entidad mostrada bajo una perspectiva particular. De esta manera, la perspectiva admite representar la variedad de presentaciones de un mismo tema de información.

Object Oriented Hypermedia Design Method o *Método de Diseño de Hipermedia Orientado a Objetos (OOHDM)*. De acuerdo con Schwabe y Rossi (1998), de la mayoría de todos los modelos propuestos para el desarrollo web que existen, es quizás el modelo OOHDM el que más ha predominado y ha sido experimentado. Esta idea surge del modelo HDM pero dentro del modelo de la orientación a objetos. Lo más atractivo del OOHDM es la gran aprobación que ha tenido el proceso de desarrollo que expone. Este paso empieza con la realización del modelo de clases conceptuales. En este, por medio de un esquema de clases, se simboliza la organización fija del sistema. Después de esto, se elabora un modelo de navegación del sistema. Este prototipo brinda una perspectiva del modelo conceptual y determina cómo se conseguirá navegar a través de la información simbolizada en el modelo conceptual. En la tercera etapa del proceso se elabora el modelo de interfaz abstracta. Este modelo entrega un panorama de cómo se va a mostrar la información al usuario. Así pues, se puede decir que el modelo de navegación es una perspectiva conceptual.

Scenario-based Object-oriented Hypermedia Design Methodology o *Método que Desarrolla Diseño en Panoramas (Scenario) Orientada a Objetos en Hipermedia (SOHDM)*. Esta es una metodología de las que dan más valor a la labor del proceso de requisitos. Se identifica principalmente porque el ciclo de vida de este empieza con la implementación de escenarios y la especificación de requerimientos (Weidenhaupt, Pohl, Jake y Haumer, 1998). El transcurso del desarrollo se segmenta en seis fases. El proceso empieza por la etapa de análisis, en la cual se lleva a cabo un estudio de los requerimientos de la aplicación, del ambiente de trabajo y de los actores. El objetivo principal de esta etapa es conseguir los escenarios que figuren las acciones que se lograrán llevar a cabo en el sistema. Después de esto se plasma un modelo de objetos en el que se despliega un diagrama de clases que simboliza la estructura conceptual del sistema. En la etapa consecutiva, el diseño de vistas, los objetos son restablecidos en elementos navegacionales que simbolizan una vista de objetos del sistema. En el ciclo del diseño navegacional se acumulan dichas vistas concretando los vínculos e hipervínculos que constan en el sistema. Y para finalizar, se contrapone a la fase de construcción en la que se realiza la aplicación.

Building Web Applications with UML o *Elaborando Aplicaciones Web con UML*. Este modelo ha obtenido gran categoría en los últimos años (Conallen, 1999). Se identifica claramente por basarse en Lenguaje Unificado de Modelado (UML) y en el proceso unificado de todo su ciclo de vida y lleva a cabo varias extensiones de estereotipos adecuados para la web, utilizando las posibilidades de prolongaciones del UML. El ciclo de vida de este modelo empieza con una etapa precursora de planificación. Después de esa etapa, inicia un proceso constante que cubre las etapas de tratamiento de requisitos, análisis, diseño implementación, pruebas y evaluación. A continuación de que el sistema haya sido aceptado, se pasa a su evaluación. Generalmente, las técnicas y los modelos que brinda son heredados de UML, pero lo beneficia y lo precisa con diversas nuevas formas de modelado específicos para la web.

Web Modelling Language o *Modelado de Lenguaje Web* (*webML*). Así como lo describieron sus autores Ceri, Fraternali y Bongio (2000), es una notación para especificar complejos web en el ámbito conceptual. Cede una representación de los sitios web desde múltiples panoramas: el conceptual, el navegacional, el de presentación, etc. El desarrollo del ciclo de vida empieza con el modelado conceptual del sistema, en el cual a través de un lenguaje de modelado UML, webML no exige ninguno determinado, se establece la organización estática del sistema. Después de esto, se lleva a cabo el modelado de hipertexto, en el cual se detallan uno o más hipertextos que obtienen ser divulgados en el sitio web. Cualquiera de estos hipertextos especifica un panorama del sitio. La especificación de los hipertextos se ejecuta a través de dos modelos: el modelo de composición, el cual precisa las páginas que forman el sistema, y el modelo de navegación, que representa cómo se obtendrá navegar a través de ellas.

Based Web Engineering o *Ingeniería Basada en la Web* (*UWE-UML*). Este modelo es una oferta fundamentada en el Proceso Unificado (Jacobson, Booch y Rumbaugh, 1999) y de igual manera con UML para el desarrollo de aplicaciones web. UWE abarca todo el ciclo de vida de esta categoría de aplicaciones, centralizando su entera aplicación en sistemas personalizados o adaptativos. Su ciclo de desarrollo se concreta en tres bases principales: la etapa de captura de requisitos, la etapa de análisis al igual que el diseño y la etapa de implementación. El periodo de desarrollo de UWE se determina por el valor que le da a la etapa siguiente, la de análisis y diseño. Todo el tiempo de desarrollo de UWE está especificado y puntualizado, así como la organización de los modelos que se van creando. No obstante, es en la fase de análisis y diseño donde se analiza más la oferta. La UWE es un planteamiento muy bien documentado que especifica perfectamente las ocupaciones a realizar. Así pues, la UWE expone las siguientes actividades:

Fase de requisitos: Propone empezar con el reconocimiento de los usuarios y la obtención de requerimientos. Emplea de distinta manera las exigencias de información, las exigencias de navegación, las exigencias de adaptación y las de interfaz de usuario, así como

algunas necesidades complementarias respectivas, por decir un ejemplo, las limitaciones de *hardware* o la de seguridad. Después de esto, concentra la atención en el estudio de los casos de uso, la elaboración de los glosarios y el arquetipo de la interfaz de usuario.

Fase de análisis y de diseño: Es semejante a la del modelo OOHDM. No obstante, abarca más características que OOHDM. Realmente UWE se diferencia por el diseño conceptual, de modelo de usuario, de navegación, de presentación, de adaptación, de la arquitectura, en el diseño detallado de las clases y en la descripción de los subsistemas e interfaces.

Fase de implementación: UWE incorpora todas las labores que trasladan a la consumación de los modelos admitidos: ejecución de la arquitectura, ejecución de la organización del hiperespacio, ejecución del modelo de usuario, ejecución de los mecanismos adaptativos y las tareas conexas a la composición de todas las implementaciones anteriores.

Basando el análisis de UWE en el procedimiento de la navegación, se puede decir que UWE expone en su fase de requisitos una adquisición de las exigencias de navegación, ya que considera estas bases de navegación como un prototipo funcional. Este estándar de navegación se elabora mediante dos etapas. En la primera fase, se expone un prototipo de espacio de la navegación, elaborado como una perspectiva del modelo conceptual y que demuestra cuáles son las clases y modelos visitables. Se constituye a través de un modelo de clases específicas llamadas clases navegacionales que no son más que clases de UML estandarizadas para demostrar su trayectoria. Este esquema se incrementa en un segundo periodo con el modelo de la estructura de la navegación.

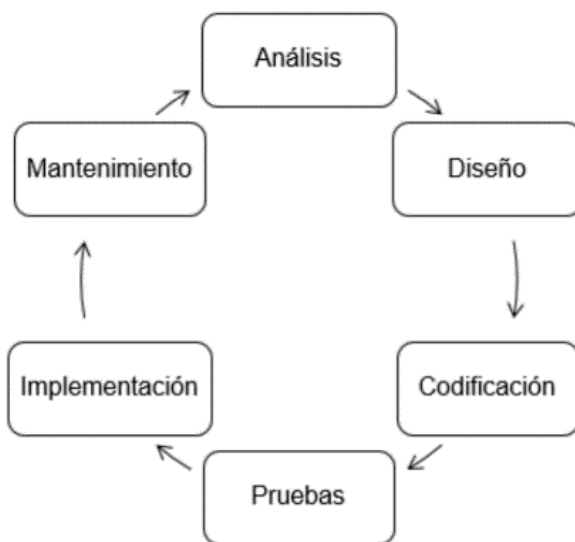
Metodología

La metodología seleccionada es la Ingeniería Basada en la Web (UWE-UML). El UWE-UML es una metodología desarrollada por Koch

(2010), del Instituto de Informática de la Universidad de Múnich en Alemania; dicha metodología proporciona un mejor panorama de una aplicación web en su fase de elaboración, dado que conserva un algoritmo apoyado en el uso del Lenguaje Unificado de Modelado (UML) para los modelos y procesos, por lo tanto, favorece su evolución. Al ser una metodología basada en UML, su proceso abarca las seis fases del desarrollo de *software* de forma cíclica. En la figura 1 se pueden apreciar las fases del modelo.

Figura 1

Fases del Modelo UWE propuesto por Koch (2010)



Dicho proceso inicia con el análisis de requerimientos del cliente, seguido del diseño del sistema y que continúa con la programación del *software* para poder realizar las pruebas necesarias y por consiguiente llevar a cabo la instalación en el servidor y así darle el mantenimiento apropiado según vayan surgiendo actualizaciones del *software* y lenguajes utilizados.

A continuación, en la tabla 1 se muestra el procedimiento para cada etapa del desarrollo del sistema web para el registro y segui-

miento a emprendedores de la incubadora de negocios Con base en la metodología del Modelo UWE-UML.

Tabla 1

Procedimiento para cada fase del desarrollo del sistema de información de inventarios Con base en la metodología del Modelo UWE-UML

Etapas	Instrumentos	Preguntas de investigación
Primera etapa: Análisis de requisitos	Instrumento de necesidades Modelo de contenido Modelo de navegación Modelo de presentación Modelo de procesos	¿Cuáles son los requisitos para el Sistema Emprendedor?
Segunda etapa: Diseño	Modelo lógico de la base de datos Modelo de la estructura web	¿Cuáles son las características del diseño del Sistema Emprendedor?
Tercera etapa: Codificación	Creación y modificación del código fuente Construcción de base de datos	¿Cuál es la codificación del Sistema Emprendedor?
Cuarta etapa: Pruebas	Pruebas de unidades Pruebas de integración Pruebas de validación	¿Qué pruebas aplicaran en el Sistema Emprendedor?
Quinta etapa: Implementación	Modelo físico de la base de datos, interfaz gráfica de la aplicación	¿Cuál será el proceso de implementación del Sistema Emprendedor?
Sexta etapa: Mantenimiento	Tipo de mantenimiento	¿Cómo será el proceso de mantenimiento del Sistema Emprendedor?

Primera etapa: Análisis de requisitos. De acuerdo con Koch (2010), en esta etapa se deben de establecer las características prácticas del sistema, como los requisitos de información, de navegación, adaptación e interfaz de usuario, así como los roles de las cuentas de los usuarios, limitaciones de *hardware* y de seguridad en el sistema web para proponerlos en un modelo de casos de uso, derivado de estos esquemas se representa como un diagrama de clases UML. A continuación, se mencionan todos los procesos siguientes: a) instrumento

de necesidades, b) modelo de contenido, c) modelo de navegación, d) modelo de presentación, y e) modelo de procesos.

Con el instrumento de necesidades se empieza precisando a los participantes del sistema y las fuentes de información para el desarrollo de este, que serán los responsables de la administración de la aplicación web. Los encargados de dicho módulo son un directivo y tres colaboradores. Planteado esto, se podrán identificar los requerimientos del sistema de información mediante un cuestionario. El estudio que se elaboró constituyó un total de diez preguntas, de las cuales siete fueron cerradas y tres fueron abiertas. Las preguntas cerradas incluyen menos de tres incisos con opción a explicar la respuesta si fuese necesario. De acuerdo con la información que se precise, algunas interrogantes solo necesitan proporcionar una respuesta de sí o no. Las preguntas abiertas poseen de una amplitud de espacio para expresar libremente la respuesta. Las preguntas de la entrevista fueron planteadas con el propósito de incluir el siguiente argumento: a) preguntas referentes a las funciones que poseerá el programa, b) preguntas respectivas a la interfaz del usuario, c) preguntas referentes al contenido que se almacenará en las tablas de la base de datos, d) preguntas respectivas a los vínculos de las tablas en la base de datos, e) preguntas referentes a los roles y seguridad que tendrán los usuarios del sistema, y f) preguntas referentes al implemento de servidores.

En lo que respecta al modelo de contenido, se precisa a través de un diagrama de clases las ideas a detalle implicadas en la aplicación. En el modelo de navegación se personaliza la navegación de los objetos que están en la aplicación y un grupo de arreglos, como son índices, menús y consultas. En el caso del modelo de presentación se incorporan las interfaces de usuario a través de vistas inconcretas. Finalmente, en el modelo de proceso se figura la vista con la que cuentan las actividades que se vinculan con cada clase de proceso.

Segunda etapa: Diseño. La segunda etapa del modelo UWE-UML, definida como “diseño”, se concentra en las especificaciones de requisitos resultantes por la fase de análisis (análisis de los requerimientos); en esta etapa se vislumbran de una manera más amplia

las vistas respectivas a los requisitos no funcionales y limitaciones con los lenguajes de programación, elementos reutilizables, sistemas operativos, tecnologías de interfaz de usuario, etc. La etapa de diseño permite el uso de tecnologías como la generación de código para la reingeniería e ingeniería inversa entre el diseño y la implementación. En esta etapa se ayuda a capturar las interfaces entre los subsistemas antes en el ciclo de vida de *software*. Además del diseño, está muy cercana a la fase de la implementación, lo que es de gran vitalidad para poder conservar y continuar el modelo de diseño durante del ciclo de vida del *software*. Esta fase de la metodología se compone de los siguientes submodelos o etapas, con los que ayuda a comprender de una manera más sencilla la importancia de la etapa del desarrollo:

1. *Modelo lógico de la base de datos*. Esta fase se separa del diseño conceptual y se elabora de modo que se adapte a la tecnología que se debe utilizar. Es indispensable que se adecue al modelo del sistema de gestión de bases de datos (SGBD) a utilizar y se obtendrán las relaciones con sus respectivos atributos, claves foranes y primarias (Costal, 2007); para lograr esto lo más factible es emplear la Teoría de la Normalización y el modelo entidad-relación, también mejor conocido como modelo E-R (Rivero, Guardia y Reig, 2004).
2. *Modelo de la estructura web*. Su propósito principal es el de personalizar de una manera visual la distribución lógica jerarquizada del sistema web a través de un esquema organizado y se debe de elaborar para plasmar la estructura de la información determinada. Este modelo es el soporte del esquema de navegación; fundamentalmente existen tres tipos de estructuras que se pueden utilizar para desarrollar nuestro sitio web, así como lo proponen Maciá y Gosende (2012).
3. *El modelo secuencial*: es la manera más práctica de organizar la información en forma de secuencia. El modelo secuencial es adecuado en sitios web de información, donde el usuario es orientado por medio de un orden de clases sucesivas a lo largo de un boceto de navegación lineal. Por ello, los algoritmos de

los buscadores web considerarán más relevantes las páginas que sean próximas a la página de inicio o “home”.

4. *Modelo jerárquico*: Este método es la mejor opción para organizar grandes cantidades de información. La mayoría de sitios web se instauran a partir de una página principal, a través de la cual se llevan a cabo los demás niveles de navegación que van de lo más común a lo más definido, por lo que esta metodología es la que más se adapta a las necesidades de organización de un sitio web.
5. *Modelo secuencial*: Este modelo requiere un sentido muy analítico, en donde se deben de establecer los distintos niveles de cada documento web. Además, beneficia a que la página principal reciba hipervínculos de las demás páginas del sitio web, ya que es de manera muy intuitiva que tenga un botón con un enlace a la página de inicio, por lo tanto, dicha página tendrá una prioridad llamada “PageAuthority”, y el algoritmo de búsqueda de los navegadores web lo podrá colocar en los resultados más destacados de Internet.
6. *El modelo en red*: Es uno de los más tolerantes en el momento que se desee organizar la información. Este tipo de modelos ayuda a tener un flujo de ideas independientes, facultando a los usuarios poder buscar sus tendencias de una manera libre. Este tipo de documentos web, en su mayoría, contienen enlaces a páginas externas, así como a las demás páginas del sitio, por ende, este tipo de modelos, además de ser útil, puede generar cierto desorden, ya que entender su comportamiento puede parecer algo complejo.

Tercera etapa: codificación. En esta fase, el analista colabora junto con los programadores para poder llevar a cabo el desarrollo del *software*. Es aquí donde se escoge el o los lenguajes de programación para poder desarrollar el proyecto y así poder convertir el diseño a un código que sea procesable para la computadora. En un proyecto, esta es la etapa más simple si los diseños de las fases anteriores fueron detallados minuciosamente, haciendo que cada módulo de la

codificación sea elaborado casi en automático. Se puede dividir en dos fases según sea requerido: creación y modificación del código fuente, y construcción de bases de datos. Estas dos fases no necesariamente existen en un mismo proyecto ya que no siempre se necesitan bases de datos para almacenar la información, como es el caso de una página web estática, la cual solo se limita a mostrar información al usuario y no necesita interactuar con sus datos; la contraparte son los sitios dinámicos, para los cuales se utiliza la programación cliente/servidor, siendo la más comúnmente utilizada PHP, como ya se vio anteriormente, para poder tener la interacción con los datos del cliente y el servidor, el cual necesita una base de datos para almacenar toda la información recabada en los campos del formulario.

Cuarta etapa: Pruebas. De acuerdo con Jacobson *et al.* (1999), en esta fase se verifican cada una de las etapas desarrolladas anteriormente, probando cada subsistema; si funcionan correctamente, se tiene la seguridad que todo lo demás funcionará sin problemas, así como las versiones que serán entregadas al usuario final. Para ello el ingeniero a cargo debe de planificar sus pruebas, por ejemplo, podría llevar a cabo un caso de uso de un subsistema y verificar que se realizó sin inconvenientes, de ser el caso contrario, revisar en qué punto de la prueba falló y realizar la ingeniería necesaria para corregirlo. Por lo anterior, se llevan a cabo las siguientes subfases, ya que son elementos fundamentales para poder ofrecer la garantía del *software*.

1. *Pruebas de unidad o de unidades:* Se hacen al finalizar la codificación de cada módulo, para verificar que funciona correctamente, es decir, que todas las especificaciones se ejecuten De acuerdo con lo establecido. Acorde con Chicano (2015), las pruebas de unidad se denominan en “caja blanca” y “caja negra”.
2. *Pruebas de integración:* Radican en verificar que la interacción entre dos o más módulos del sistema funcione correctamente. Es de vital importancia mencionar que antes de realizar estas pruebas, se deben de verificar los módulos individualmente (Cardador, 2014).

3. *Pruebas de validación.* Se llevan a cabo con la tarea de constatar si se está elaborando el producto apropiado (Garzón, Sampalo de la Torre, Leyva y Prieto, 2004). La validación asegura que el sistema desarrollado cumpla con los requerimientos que planteó el cliente. El autor Fernández (2006) señaló que la prueba de validación se divide en: rendimiento del sistema, rendimiento del proceso durante sobrecarga de trabajo, prueba de ergonomía, prueba de métodos y procedimientos, prueba de copias de seguridad y recuperación.

Quinta etapa: Implementación. De acuerdo con Sommerville y Alfonso (2005), la fase de implementación es el procedimiento de transformar un requerimiento del sistema en un sistema ejecutable, siempre tomando en cuenta las fases de diseño y codificación del *software* y, siendo necesario, también se pueden tomar medidas para hacer una mejora de los requerimientos del sistema. De igual manera, Barraza (2010) mencionó que la implementación es el proceso de elaborar los módulos de *software* indispensables para evolucionar los diseños de datos e hipertexto en una aplicación que se pueda ejecutar en la plataforma establecida. La implementación se divide en dos subfases: modelo físico de la base de datos o implementación de datos, e interfaz gráfica de la aplicación o implementación de hipertexto.

Sexta etapa: Mantenimiento. De acuerdo con Joyanes y Angoar (1997), esta última etapa es una de las más implementadas en el ciclo de vida de un *software*, es una fase en donde se mejoran los procesos y se optimiza el *software* una vez que haya sido entregada la revisión del programa al usuario final. La etapa del mantenimiento del *software* implica cambios al programa para reparar inconvenientes encontrados durante su empleo o durante el anexo de nuevas funciones. Existen errores que son descubiertos por los usuarios del *software* al momento de utilizarlo, los cuales son reportados al equipo de desarrollo para su pronto rastreo. Para llevar a cabo un

buen mantenimiento del *software* se describen cuatro tipos, los cuales se describen enseguida:

1. *Perfectivo*: Este tipo de mantenimiento se implementa para mejorar el sistema debido a nuevos inconvenientes en cualquiera de sus diferentes aspectos, como son modificación del código, un nuevo concepto en el sistema, optimización del rendimiento y eficiencia.
2. *Evolutivo*: Son los anexos, las modificaciones y eliminaciones indispensables en el *software* para abarcar el crecimiento o cambio en las necesidades del usuario.
3. *Adaptativo*: Son los cambios que afectan a los entornos en los que labora el sistema, por ejemplo: cambios en los requerimientos del *hardware*, cambio de gestores de base de datos, y otros.
4. *Correctivo*: Son los cambios que son necesarios para corregir los errores del producto final.

Resultados

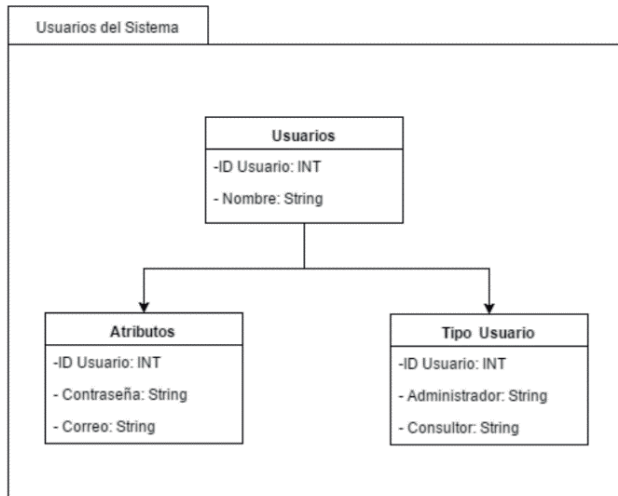
Los resultados se presentan Con base en las seis fases del modelo UWE-UML propuesto por Koch (2010).

Fase 1. Análisis. En esta fase se describen a detalle los resultados obtenidos de los instrumentos utilizados para la fase de análisis de los requerimientos del Sistema Emprendedor. Con base en los resultados del instrumento de diagnóstico, el cual fue aplicado a los directivos de la empresa Vallarta Emprende, los requisitos del Sistema Emprendedor fueron los siguientes:

1. Modelos de contenido, para los diferentes escenarios que se encontraron después de aplicar el análisis de requisitos; en la figura 2 se muestra un ejemplo del modelo de contenidos para el Sistema Emprendedor.

Figura 2

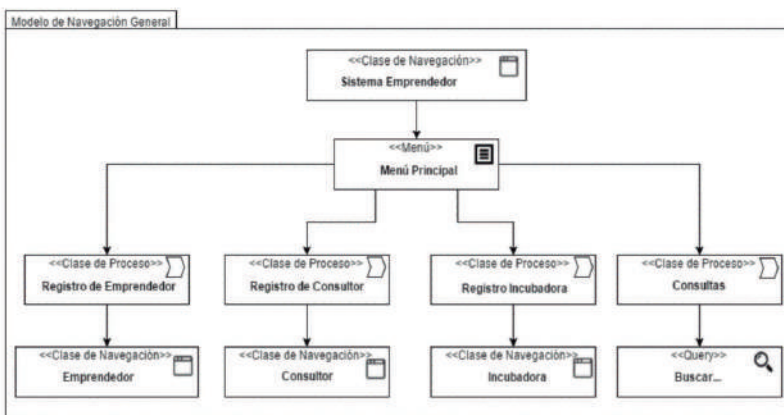
Modelo de contenidos para usuarios en el Sistema Emprendedor



2. Modelo de navegación; en la figura 3 se muestra un ejemplo del modelo de navegación del usuario del Sistema Emprendedor.

Figura 3

Modelo de navegación del usuario para el Sistema Emprendedor



3. Modelo de presentación; en la figura 4 se muestra un ejemplo de la presentación de inicio de sesión del Sistema Emprendedor.

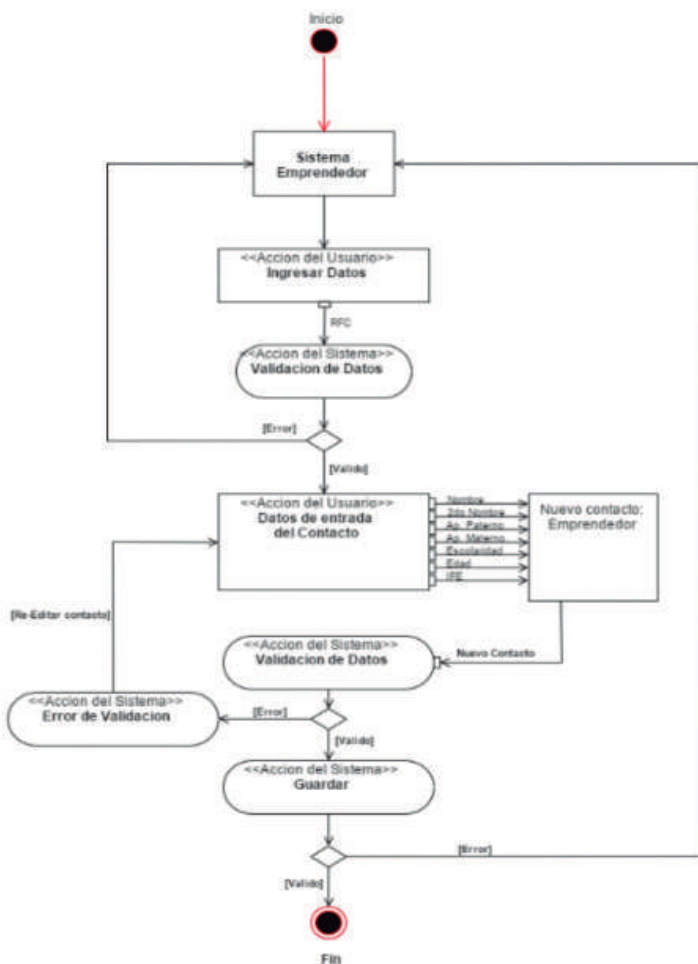
Figura 4

Modelo de presentación del usuario para el Sistema Emprendedor



4. Modelo de procesos; en la figura 5 se muestra un ejemplo de procesos del Sistema Emprendedor.

Figura 5
Modelo de procesos del Sistema Emprendedor

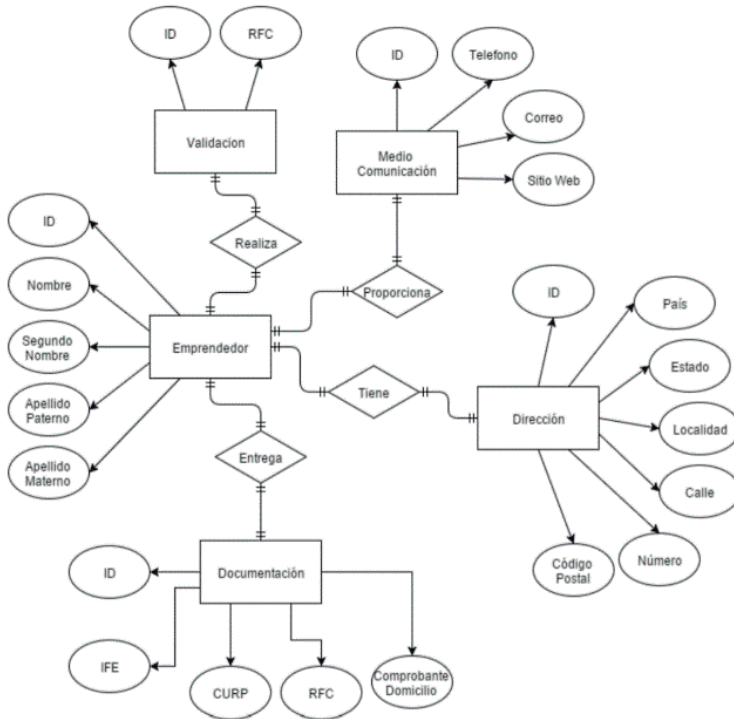


Fase 2. Diseño. En esta etapa se desarrolló el modelo lógico de la base de datos del Sistema Emprendedor, la cual se concretó con el modelo entidad-relación (E-R). A continuación se presenta un ejemplo de los modelos desarrollados.

5. Modelo lógico de la base de datos; en la figura 6 se muestra el esquema lógico para emprendedores del Sistema Emprendedor.

Figura 6

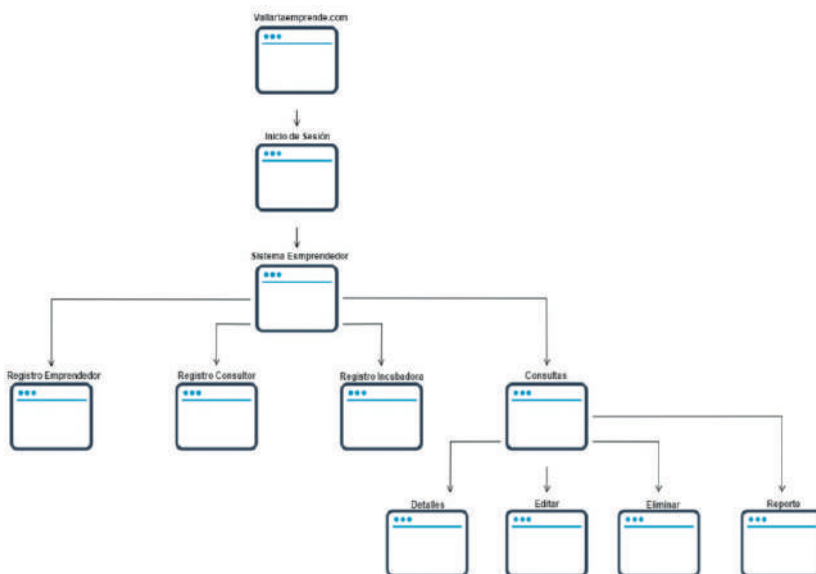
Esquema lógico para emprendedores del Sistema Emprendedor



6. Modelo de estructura web; en la figura 7 se muestra el modelo de la estructura jerárquica del Sistema Emprendedor.

Figura 7

Modelo de estructura jerárquica del Sistema Emprendedor



Fase 3. Codificación, creación y modificación del código fuente. En este apartado se muestran los resultados de la codificación realizada para el Sistema Emprendedor. En la figura 8 se muestra el fragmento de código HTML del menú principal del Sistema Emprendedor.

Figura 8

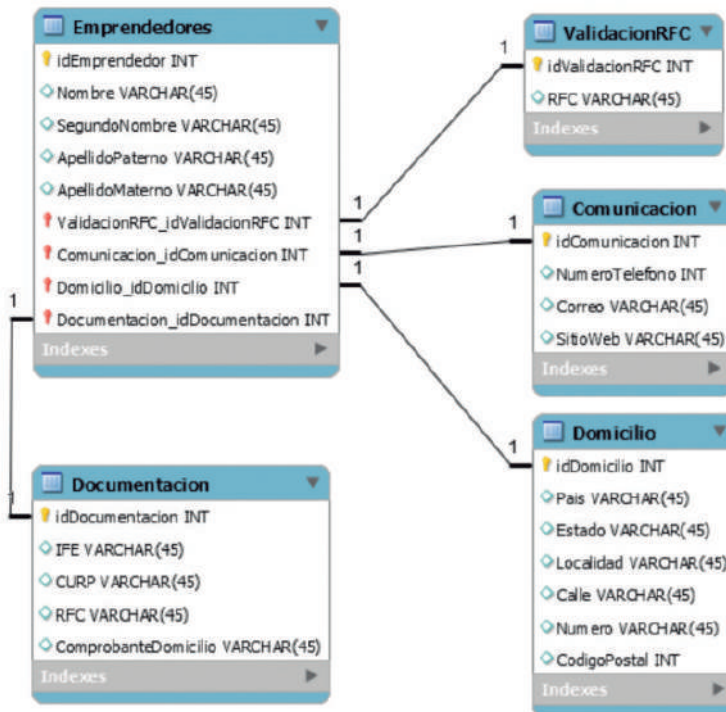
Fragmento de código HTML del menú principal del sistema del Sistema Emprendedor

```
20 <h1>Sistema Emprendedor</h1>
34 </div>
37 <div class="row">
38 <div class="col-lg-3 col-md-4 col-xs-6 thumb">
39 <a class="thumbnail" href="emprendedor/rfc.html">
40 
41 </a>
42 <h3>Registrar Emprendedor</h3>
43 </div>
44 <div class="col-lg-3 col-md-4 col-xs-6 thumb">
45 <a class="thumbnail" href="consultor/">
46 
47 </a>
48 <center><h3>Registrar Consultor</h3></center>
49 </div>
50 <div class="col-lg-3 col-md-4 col-xs-6 thumb">
51 <a class="thumbnail" href="incubadora/">
52 
53 </a>
54 <center><h3>Registrar Incubadora</h3></center>
55 </div>
56 <div class="col-lg-3 col-md-4 col-xs-6 thumb">
57 <a class="thumbnail" href="consultas/emprendedor">
58 
59 </a>
60 <center><h3>Consultas</h3></center>
```

Asimismo, como parte de los resultados de esta fase se muestra la interfaz gráfica de la creación de la base de datos del Sistema Emprendedor. En la figura 9 se muestra el esquema general de las tablas de registro de un emprendedor del Sistema Emprendedor.

Figura 9

Esquema general de las tablas de registro de un emprendedor del Sistema Emprendedor



Fase 4. Pruebas de unidad. En la figura 10 se muestra la prueba de caja blanca del menú principal del Sistema Emprendedor.

Figura 10

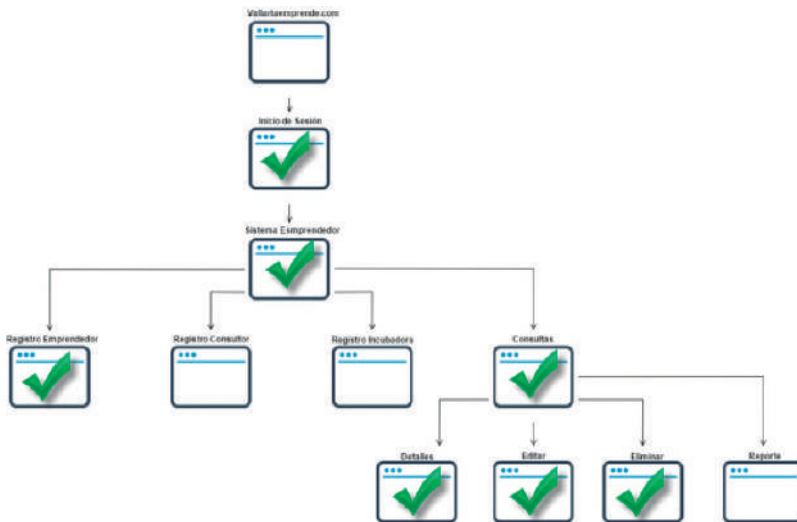
Prueba de caja blanca del menú principal del Sistema Emprendedor



A continuación, se muestra la prueba de integración del Sistema Emprendedor. En la figura 11 se muestra la prueba no incremental del Sistema Emprendedor.

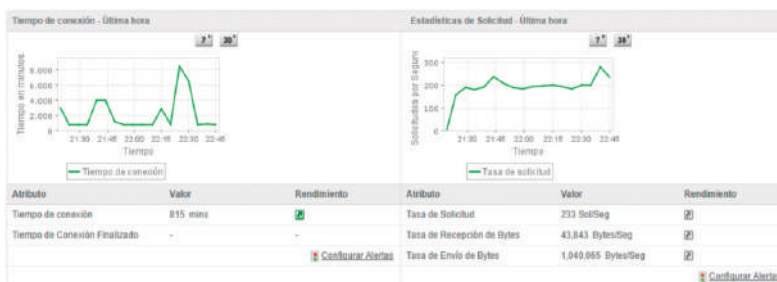
Figura 11

Prueba no incremental del Sistema Emprendedor



A continuación se muestran las pruebas de validación aplicadas al Sistema Emprendedor. En la figura 12 se muestra la prueba de rendimiento del sistema.

Figura 12
Prueba de rendimiento del Sistema Emprendedor



Fase 5. Implementación. En este apartado se muestra el modelo físico del Sistema Emprendedor. En la figura 13 se muestra el diseño físico de tabla de emprendedores del Sistema Emprendedor.

Figura 13
Diseño físico de tabla de emprendedores del Sistema Emprendedor

The screenshot shows a table structure view for a database. The table 'emprendedores' is highlighted, showing its columns and data types.

Tabla	Replicación	Acción	Columna	Tipo	Cotejamiento	Tamaño	Residuo a depurar
comunicacion		Examinar Estructura Buscar Insertar Vaciar Eliminar	id	InnoDB	utf8_spanish2_ci	16 KB	-
documentacion		Examinar Estructura Buscar Insertar Vaciar Eliminar	id	InnoDB	utf8_spanish2_ci	16 KB	-
domicilio		Examinar Estructura Buscar Insertar Vaciar Eliminar	id	InnoDB	utf8_spanish2_ci	16 KB	-
emprendedores		Examinar Estructura Buscar Insertar Vaciar Eliminar	id	InnoDB	utf8_spanish2_ci	16 KB	-
validacionrfc		Examinar Estructura Buscar Insertar Vaciar Eliminar	id	InnoDB	utf8_spanish2_ci	16 KB	-
6 tablas		Replicación	Número de filas	InnoDB	utf8_spanish2_ci	368 KB	0 B

En cuanto a la interfaz gráfica de la aplicación desarrollada del Sistema Emprendedor, en la figura 14 se muestra el menú principal.

Figura 14
Interfaz gráfica de la aplicación desarrollada del Sistema Emprendedor



Fase 5. Implementación. De acuerdo con los autores Joyanes y Angoar (1997), existen cuatro tipos de mantenimientos: a) perfecto, b) evolutivo, c) adaptativo, y d) correctivo. Con base en lo anterior, el tipo de mantenimiento que se propone a implementar en el Sistema Emprendedor es el mantenimiento evolutivo, ya que su definición se adecua a las necesidades a futuro del sistema, como son: los anexos de nuevos módulos, por ejemplo, un gestor de cuentas y perfiles de usuarios. Adicionalmente, se plantea realizar un mantenimiento correctivo durante su primer año de implementación, ya que por ser la versión inicial del *software* se podrían encontrar inconsistencias en el sistema. Siendo así, una vez terminada esta etapa de su ejecución, se propuso realizar un mantenimiento perfecto cada cinco años, que es la vida útil de un *software* (Berumen y Arriaza, 2008).

Discusión y conclusiones

La concreción del Sistema Emprendedor permitió dar solución a las problemáticas de las necesidades detectadas en la incubadora de negocios. Este sistema permitió lo siguiente: a) facilitar su acceso debido a que es basado en Web, b) eficientar los procesos de registros de los emprendedores, c) tener centralizada la información de manera segura, y d) innovar los procesos realizados con apoyo de

las tecnologías de la información. Básicamente, el Sistema Emprendedor cumplió con los requerimientos solicitados por las personas que trabajan en la incubadora de negocios, logrando evidenciar su funcionamiento de manera efectiva.

Este proyecto puede ser mejorado en diferentes aspectos. Primero, cambiar el diseño de la interfaz del Sistema Emprendedor, para hacerlo más agradable al usuario que lo opera. Segundo, implementar el *framework Bootstrap*, el cual ayuda a visualizar el contenido de una página web en cualquier tipo de resolución de los dispositivos, facilitando así el acceso al Sistema Emprendedor desde dispositivos móviles. Tercero, agregar un panel de control al Sistema Emprendedor, donde el administrador del sistema cuente con las herramientas necesarias para ejercer su cometido sin problemas, como, por ejemplo, el registro de nuevos usuarios, asignación de roles, administración de perfiles de usuarios y otros. Cuarto, hacer un nuevo modelo de la base de datos donde se agregarán las tablas necesarias para registrar la información de los nuevos módulos del Sistema Emprendedor. Quinto, crear una base de datos en un nuevo servidor, ya sea externo o local, donde se replicarán de manera automática todos los registros de la base de datos principal “vallartaemprende”, teniendo así una copia de seguridad. Sexto, aplicar todas las validaciones de los campos en los formularios del Sistema Emprendedor, para optimizar los registros y así evitar anomalías en la base de datos. Finalmente, revisar que los requerimientos sigan cumpliendo las expectativas que se propusieron al principio del proyecto.

Referencias

- Barraza, F. (2010), *Metodologías de diseño de aplicaciones web, Parte B*, Colombia, Calí, Pontificia Universidad Javeriana.
- Berumen, S. y K. Arriaza Ibarra (2008), *Evolución y desarrollo de las TIC en la economía del conocimiento*, ECOBOOK.

- Cancelo López, P. y J. M. Alonso Giráldez (2007), *La Tercera Revolución*, España, NetBiblio.
- Cardador Cabello, A. L. (2014), “Implementación de aplicaciones web en entornos Internet, Intranet y Extranet”, en S. Berumen y K. Arriaza Ibarra (2008), *Evolución y desarrollo de las TIC en la economía del conocimiento*, Málaga, ECOBOOK, IC Editorial.
- Ceri, S.; Fraternali, P. y A. Bongio (2000), *Web Modelling Language (webML): A Modelling Language for Designing Web Sites*, Italia, Politécnico di Milano.
- Chen, P. (1976), *The Entity-relationship Approach: Towards a Unified Behavior of Data*, EE.UU., Nueva York, ACM Transactions on Database Systems.
- Chicano Tejada, E. (2015), *Auditoría de seguridad informática*, IFCT0109, España, IC Editorial.
- Cobo, Á.; Gómez, P. y R. Rocha (2005), *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*, España, Diaz de Santos.
- Cobo, A. (2007), *Diseño y programación de base de datos*, España, Madrid, Visión Libros.
- Conallen, J. (1999), *Building Web Applications with UML*, California, Cupertino, Rational.
- Costal Costa, D. (2007), “Introducción al diseño de bases de datos”, en *Bases de datos*, pp. 8-10, España, UOC.
- Deshpande, Y.; Murugesan, S.; Ginige, A.; Hansen, S.; Schwabe, D.; Gaedke, M. & B. White (2002), *Web Engineering Journal of Web Engineering*, Nueva Jersey, Princeton, Rinton Press.
- Fernández Alarcón, V. (2006), “Una metodología basada en el modelado”, en V. F. Alarcón, *Desarrollo de sistemas de información*, p. 11, España, Barcelona, Edicions UPC.
- Garzón Villar, M. L.; Sampalo de la Torre, M.; Leyva Cortés, E. e I. Prieto Tinoco (2004), *Informática. Temario A. Volumen II, Profesores de Educación Secundaria*, España, Editorial MAD.
- González Castellanos, E. (2010), *En la computadora personal y sus conceptos básicos*, p. 81, Puerto Rico, Advanced Graphic Printing.
- INADEM (2015), *INADEM*. Disponible en: https://www.inadem.gob.mx/templates/protostar/que_es_inadem.php
- Jacobson, I.; Booch, G. y J. Rumbaugh (1999), *The Unified Software Development Process*, Columbia, Addison-Wesley Professional.

- Joyanes Aguilar, L. y M. Angoar (1997), *Ingeniería del software: un enfoque práctico*. Disponible en: [https://books.google.com.mx/books?id=8U-v5jxkuBZ1C&printsec=frontcover &source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false](https://books.google.com.mx/books?id=8U-v5jxkuBZ1C&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Jiménez Capel, M. Y. (2014), *Bases de datos relacionales y modelado de datos*, IFCT0310, España, Málaga, IC Editorial.
- Koch, N. (2010), *UML-based Web Engineering*, Almería, LMU.
- Laudon, K. C. y J. P. Laudon (2016), *Sistemas de información gerencial*, México, Pearson Educación.
- López Herrera, J. (2011), *Programación en tiempo real y bases de datos: un enfoque práctico*, Catalunya, UPCGRAU.
- Luján Mora, S. (2002), *Programación de aplicaciones web: historia, principios básicos y clientes web*, Alicante, ECU.
- Luke, A. (2015), “Las ventajas de las bases de datos basadas en Web”, *La Voz de Houston*. Disponible en: <http://pyme.lavoztx.com/las-ventajas-de-las-bases-dedatos-basadas-en-10686.html>
- Maciá Domene, F. y J. Gosende Grel (2012), *Posicionamiento en buscadores*, Madrid, Anaya Multimedia.
- Peña Ayala, A. (2006), *Ingeniería de software: una guía para crear sistemas de información*, México D.F., Instituto Politécnico Nacional.
- Pérez de Ovalles, M. A. y L. E. Mendoza M. (2014), *La reingeniería de los procesos del negocio*, Universidad Simón Bolívar.
- Piñeiro Gómez, J. M. (2013), *Bases de datos y modelado de datos*, España, Madrid, Paraninfo.
- (2014), *Diseño de bases de datos relacionales*, España, Paraninfo.
- Pons, O.; Marín, N.; Medina, J. M.; Acid, S. y M. A. Vila (2009), *Introducción a las bases de datos*, España, Madrid, Thomson.
- Real Academia Española (2012), “Sistema”, octubre. Disponible en: <http://lema.rae.es/drae/?val=sistema>
- Resino Minayo, C. y B. Ena Ventura (2007), “Gestión administrativa”, en *Aplicaciones informáticas*, p. 319, España: Thomson Paraninfo.
- Rivero Cornelio, E.; Guardia Rivas, C. y J. C. Reig Hernández (2004), “Bases de datos relacionales”, en *Diseño físico: orientada al DB2 para z/OS de IBM*, España, Universidad Pontificia Comillas.

- Salvador, S. G. (2003), “Ingeniería de proyectos informáticos”, en *Ingeniería de proyectos informáticos: actividades y procedimientos*, pp. 5-7, Universidad Jaume I., Servicio de Comunicación y Publicaciones.
- Schwabe, D. y G. Rossi (1998), *Developing Hypermedia Application*, EE.UU., Pittsburgh, Springer.
- Sommerville, I. y M. I. Alfonso Galipienso (2005), *Ingeniería del software*, 7ª ed., España, Madrid, Pearson Educación, S.A.
- Stair, R. y G. Reynolds (2010), *Principios de sistemas de información: un enfoque administrativo*, México D.F., CENGAGE Learning.
- Vallarta Emprende (2015), “Vallarta Emprende”. Disponible en: <http://vallartaemprende.com/>
- Weidenhaupt, K.; Pohl, K.; Jake, M. y P. Haumer (1998), “Scenarios in System Development: Current Practice”, *IEEE Software*.
- W3C (2015), “W3C”. Disponible en: <http://www.w3c.es/>

3

APLICACIÓN MÓVIL Y SISTEMA DE SOPORTE A LA DECISIÓN PARA LA OPTIMIZACIÓN DE SOLUCIÓN DE REPORTES EN UNA FACULTAD UNIVERSITARIA*

YOBANI MARTÍNEZ RAMÍREZ¹

ALAN RAMÍREZ NORIEGA²

JOSÉ COTA COTA³

MARTÍN SÁNCHEZ MUNDO⁴

Resumen

Las universidades inteligentes son aquellas que hacen uso de la tecnología, el *software* y la Internet de las cosas (IoT) para realizar un análisis del entorno y conocer la situación en la que se encuentra. Dentro de las facultades académicas de una universidad pueden iden-

* Derivado del proyecto de investigación: "Actividades de Investigación del Cuerpo Académico uas-ca-295".

1 Universidad Autónoma de Sinaloa, correo electrónico: yobani@uas.edu.mx

2 Universidad Autónoma de Sinaloa, correo electrónico: alandramireznoriega@uas.edu.mx

3 Universidad Autónoma de Sinaloa, correo electrónico: eduardocc.8.20@gmail.com

4 Universidad Autónoma de Sinaloa, correo electrónico: kdvn_007@hotmail.com

tificarse distintos tipos de problemas administrativos que, debido al tamaño de la institución, puede demorar su solución ya que el problema puede tardar en ser comunicado a las autoridades correspondientes. En este trabajo se propone desarrollar una aplicación móvil que permita comunicar los problemas y las situaciones que suceden dentro de una facultad al personal administrativo de la institución. Al mismo tiempo, se propone un sistema de soporte a las decisiones (SSD) sobre plataforma web (aplicación web), el cual permita tanto la gestión de los problemas administrativos como la explotación de la información generada para la toma de decisiones eficaces. Se realizó una evaluación que duró tres semanas con un grupo experimental conformado por 26 estudiantes y un administrador de una facultad universitaria. Se aplicaron ocho preguntas de la prueba de usabilidad Software Usability Measurement Inventory (SUMI) para conocer el punto de vista de los usuarios con respecto a la calidad de uso. El análisis de los resultados determinó que ambos sistemas son usables. Se concluye que el sistema en general (web y móvil) es estable, tanto en su funcionamiento como su usabilidad.

Palabras clave: universidad inteligente, aplicación móvil, usabilidad.

Abstract

Smart universities are those that make use of technology, software, and the Internet of Things (IoT) to carry out an analysis of the environment and know the situation in which it finds itself. Different types of administrative problems can be identified in universities. These problems may be delayed in their solution due to the corresponding management. In this work, it is proposed to develop a mobile application that allows communicating the problems and situations that occur within a school to the administrative staff of the institution. At the same time, a Decision Support System (SSD) is proposed on a web platform (web application). In such a way that

the application allows both the management of administrative problems and the exploitation of the information generated for effective decision making. An evaluation was carried out with an experimental group made up of 26 students and an administrator from a university faculty. Questions from the SUMI (Software Usability Measurement Inventory) usability test were applied to find out the users' point of view regarding the quality of use. The analysis of the results determined that both systems (web and mobile) are usable and functional.

Keywords: smart university, mobile application, usability.

Introducción

En la actualidad, con la implementación del Internet de las Cosas (IoT) y las tecnologías de información y comunicación (TIC), se comenzaron a crear propuestas en las que los ciudadanos podrían comunicar de una forma más rápida a las instituciones de gobierno sobre los problemas o inconformidades dentro de la ciudad, dando el nombre a este concepto de “ciudad inteligente” (Parra-Valeancia, Guerrero y Rico, 2017).

El concepto de ciudad inteligente se ha ido trasladando a un entorno universitario puesto que lo que busca solucionar una ciudad inteligente es el mismo sistema de reporte de problemas y quejas que se maneja en una facultad universitaria (Maciá Pérez, 2017). Por medio del uso de las tecnologías móviles es posible crear enlaces entre el personal administrativo y los estudiantes, donde un estudiante o docente, por medio de una aplicación móvil, pueda crear el reporte de un problema y de forma inmediata sea este recibido por los encargados de dar solución a los distintos problemas que pueden presentarse (infraestructura, mantenimiento, limpieza, zonas de riesgo, conflictos o situaciones no deseables, entre otros). Para ofrecer flexibilidad en cuanto al análisis de los problemas por parte del área administrativa, estos problemas pueden ser visualizados y

gestionados a través de una aplicación web, reduciendo las limitantes de ubicación en las que puede ser utilizada esta herramienta.

En ocasiones, la administración de las facultades de la universidad no cuenta con un medio eficiente por el cual se puedan reportar los problemas administrativos, por lo que muchos de estos son omitidos ya que los estudiantes no consideran necesario hacer el proceso para reportar un problema.

Por otro lado, aun cuando los problemas son reportados y solucionados, no se mantiene una retroalimentación con la persona que hizo el reporte, por lo que no hay forma de que los problemas (resueltos o en seguimiento) sean informados a los interesados. Además, debido a esta falta de seguimiento solo se genera información en bruto y dificulta tomar decisiones a una escala más amplia, es decir, tomar decisiones a partir de un conjunto de problemas reportados; esto provoca soluciones en un periodo de tiempo más largo y no eficientes.

La implementación de “universidades inteligentes” utilizando aplicaciones móviles y aplicaciones web permite acceder a información estructurada; estos datos pueden ser tratados mediante algoritmos de soporte a las decisiones con el fin de generar nuevo conocimiento que aumente la eficiencia de la toma de decisiones en la resolución de problemas dentro de la facultad universitaria, derivando de esto una mejor oferta de recursos a los estudiantes e interesados.

Revisión de la literatura

En este apartado se abordan las investigaciones realizadas por diversos autores y que se relacionan con este proyecto de investigación. Se mencionan algunas de las aportaciones más relevantes que influyen en este trabajo, hablando no solo de la parte tecnológica, sino del concepto de este tipo de aplicaciones. Dando un contraste desde aplicaciones orientadas a las ciudades inteligentes hasta evolucionar

a lo que se llama “*Smart University*” o universidades inteligentes y “*Smart Citizen*” o ciudadanos inteligentes.

Los autores Desouza y Bhagwatwar (2012) realizaron un análisis de distintas aplicaciones que permiten la retroalimentación ciudadana a las organizaciones capaces de dar solución a los problemas reportados, identificando distintos tipos de aplicaciones, siendo estas divididas por la meta de la aplicación, la motivación que existe detrás, la naturaleza de la información y los retos que apunta a resolver. El resultado del análisis trae consigo un conjunto de conceptos que permiten tratar de manera más clara la idea de una persona como retroalimentador para la solución de problemas en un área social.

En la investigación desarrollada por Lemos y Veras (2018) se realizó un conjunto de encuestas para conocer más a fondo el uso de aplicaciones para el reporte de problemas. Estas encuestas señalan la interacción de las funcionalidades de la aplicación con los usuarios, dando como resultado el comportamiento de los usuarios según sus necesidades, así como también el nivel de uso de la aplicación. También brinda conceptos que permiten entender la estructura de este tipo de aplicaciones, como es, principalmente, la definición de ciudad inteligente a partir de ciudadanos como sensores. El ciudadano como sensor hace referencia a la forma como funcionan los sensores digitales que captan información haciendo lecturas y traducciones sobre las magnitudes del medio ambiente (temperatura, ruido, CO₂ en datos digitales, por ejemplo), y luego envían la información para la toma de decisión en otros sistemas.

El autor Chichernea (2014) realizó un análisis de las características de los sistemas de soporte a las decisiones (SSD) orientados a la administración de una ciudad. La investigación obtiene como resultado algunas de las ventajas y desventajas que trae el uso de esta clase de sistemas en la administración ciudadana sin descartar las circunstancias en las que se implementa. Describe los elementos con los que trabaja un SSD definiendo el proceso que ocurre para convertir un conjunto de datos en información que puede derivar en acciones.

El autor Balakrishna (2012) mostró una investigación sobre los conceptos de “ciudad inteligente” y “sensor inteligente”, este último haciendo referencia a las personas como perceptores de lo que sucede a su alrededor. La investigación brinda como resultado una visión de cómo podrían evolucionar dichos conceptos en un futuro considerando los avances en el principal medio que son las tecnologías, tales como los dispositivos móviles y la forma en la que estos pueden generar información de lo que está sucediendo en su entorno. Uno de los principales resultados de la investigación de Balakrishna es el relacionado con la escalabilidad de las aplicaciones.

Los autores Parra-Valeancia, Guerrero y Rico (2017) hacen un análisis de la Internet de las cosas (IoT, en inglés “Internet of Things”), como la conexión de objetos tecnológicos o electrónicos hacia Internet. Este concepto se deriva del avance de la tecnología y de la necesidad de compartir y controlar las cosas que nos rodean. Según los autores, la IoT es la principal unión entre los conceptos de *Smart City* y *Smart University*. En cualquier caso, una ciudad inteligente es esencialmente habilitada por el uso de tecnologías para mejorar la competitividad y garantizar un futuro más sostenible. Más concretamente, las estrategias e iniciativas de una *Smart City* deben incluir al menos una de las siguientes características: a) *Smart Governance*, b) *Smart People*, c) *Smart Living*, d) *Smart Mobility*, e) *Smart Economy*, f) *Smart Environment*.

Se puede asumir que una ciudad inteligente es un territorio caracterizado por el uso intensivo de las tecnologías, principalmente de información y comunicación, para promover la colaboración, la innovación y la eficiencia para lograr el mejoramiento de la calidad de vida de los ciudadanos y la sostenibilidad de las ciudades a través del mejoramiento de los servicios hacia los ciudadanos.

Ahora, partiendo de este concepto de *Smart City*, trasladado a toda su comunidad y a su entorno, generan como resultado el concepto de *Smart Campus* o *Smart University*, el cual mantiene el mismo objetivo principal de mejorar la calidad de vida de su comunidad aplicando de forma global, intensiva y sostenible las TIC bajo el principio de servicio

a la ciudadanía. Para lograr todo es necesario apoyarse en el poder de las TIC que existen actualmente, ya que su correcto uso será el pilar fundamental para el éxito fundamental del proyecto.

Material y métodos

El desarrollo de un *software* incluye las etapas genéricas de análisis, diseño, implementación y prueba Pressman (2010), estas etapas bien aplicadas garantizan un *software* desarrollado De acuerdo con los requerimientos. A continuación, se describe cada una de las etapas mencionadas.

A) *Análisis del sistema*

La tarea del análisis es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del *software* y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo (Pressman, 2010). El análisis realizado al planteamiento del problema requiere el desarrollo de un sistema dividido en dos partes: una aplicación móvil y una aplicación web (que facilite la toma de decisiones) enfocados al reporte de problemas para dar una pronta solución. Para apoyar en la tarea de toma de decisiones se decidió utilizar el algoritmo Apriori desde el servidor web.

El algoritmo Apriori es un algoritmo utilizado en la minería de datos sobre bases de datos transaccionales, el cual sirve para generar reglas de asociación. Apriori emplea una búsqueda de arriba hacia abajo que enumera cada uno los conjuntos de datos frecuentes. La complejidad de este algoritmo restringe su uso en conjuntos de datos pequeños o patrones cortos (Bayardo, 1998).

B) Diseño del sistema

El diseño es la primera de las tres actividades técnicas que implica un proceso de ingeniería de *software*; estas etapas son diseño, codificación y pruebas. Generalmente la fase de diseño produce un diseño de datos y un diseño arquitectónico (Pressman, 2010). En este trabajo de investigación, el diseño de datos juega un papel muy importante, ya que se planea explotar la información por medio de un sistema de soporte a las decisiones (SSD), por ello los datos generados por la aplicación móvil deben ser consistentes y precisos. Por otra parte, el diseño arquitectónico cumple un papel menos importante que el diseño de datos, pero no deja de ser relevante ya que, al ser un sistema compuesto por dos aplicaciones, es importante que la estructura de cada uno de estos módulos esté definida correctamente.

b.1. Arquitectura

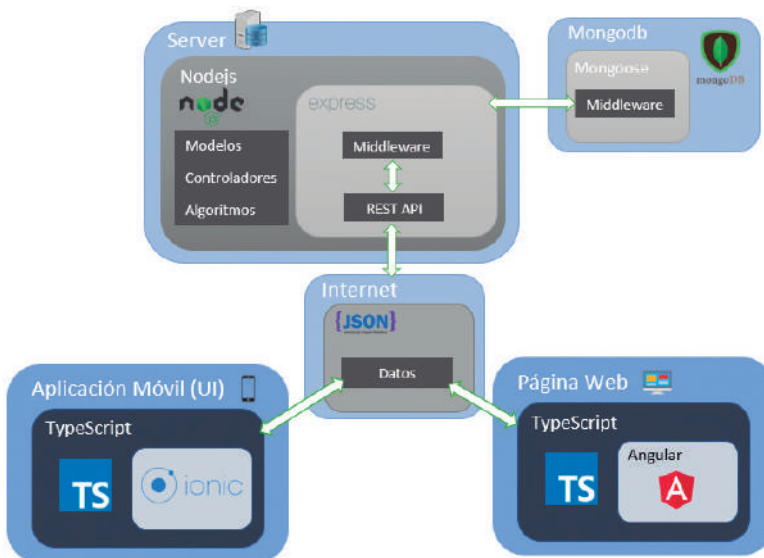
La arquitectura en capas es una de la más fuertemente adoptadas por los desarrolladores. Una capa representa un aspecto lógico a tratar como una unidad propia con sus objetivos funcionales claramente definidos, sus dependencias y sus colaboraciones hacia el sistema (Pressman, 2010). Para explicar la arquitectura del *software* desarrollado, se puede pensar en un sistema que ofrece las funcionalidades básicas CRUD (cuyas siglas significan Crear, Leer, Actualizar y Eliminar). Al analizar las actividades de un CRUD, cada una de ellas podría involucrar tres tipos de funcionalidades distintas:

- *Las relacionadas con el usuario.* La manera en que el sistema interactúa con él, la captura de los datos que ingresa y su visualización como resultado de consultas o cualquier otra operación que implique la recuperación de datos.
- *Las reglas lógicas.* Son las que afectan a los datos que el usuario ha ingresado generando nueva información a partir de ellos. También se incluyen los datos calculados sobre los datos almacenados, cuando existe un requerimiento de visualización.

- *La persistencia de los datos.* Se refiere al tratamiento de los estados de los objetos para ser almacenados físicamente en la base de datos.

Considerando esto, en la figura 1 se plantea una arquitectura de tres capas. Esta arquitectura representa la manera en que el *software* interactúa entre los distintos módulos para lograr los objetivos definidos. Esta modularización sirve para dividir el trabajo de programación en partes más pequeñas; cada una de estas partes debe ser lo más independiente del resto, logrando con esto reducir la complejidad del desarrollo (Sommerville, 2005).

Figura 1
Arquitectura del sistema



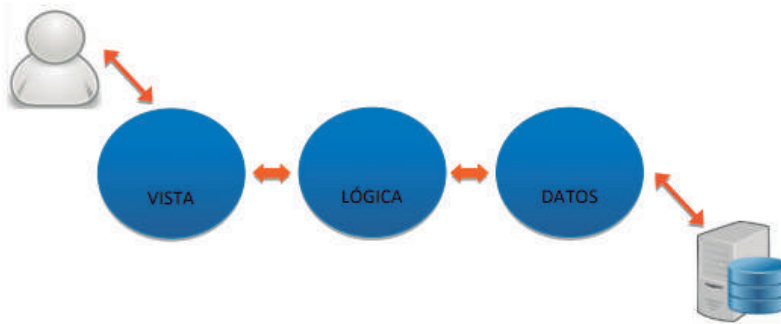
Fuente: Elaboración propia.

La figura 1 presenta la estructura de la arquitectura, que está dividida en tres capas: capa de presentación, capa lógica de negocios y capa

de datos orientada a una arquitectura cliente/servidor. El esquema de la arquitectura se muestra en la figura 2.

Figura 2

Esquema de una arquitectura de tres capas



Fuente: Elaboración propia.

Capa de presentación

En ella se colocará todo aquello que permite controlar los aspectos que estén relacionados con la interfaz gráfica del usuario, así como lo necesario para interpretar las acciones que este solicite y las formas de mostrarle la información. Por ejemplo, si el usuario solicita grabar los datos ingresados en la interfaz gráfica de usuario (GUI), este objeto sería el encargado de traducir esa petición en un mensaje concreto al objeto que corresponda de la capa de lógica, además de proveerle la referencia al objeto que posee el estado a persistir. Si por el contrario la solicitud del usuario es una consulta, la capa de presentación transformará este requerimiento en un mensaje al objeto indicado de la capa de lógica, conjuntamente con una referencia al objeto que posee el código a consultar y permanecerá (esto también puede construirse de forma asincrónica) a la espera del arribo de los datos consultados para presentarlos adecuadamente en la GUI.

La capa de presentación está dividida en dos aplicaciones, sin embargo, para optimizar el desarrollo de ambas partes, se decidió

un marco de trabajo el cual facilite la programación, lo cual permitiría una homogeneidad de lenguajes reduciendo la complejidad. Para la capa de presentación de la aplicación móvil se desarrollará una PWA (Progressive Web Application) con el *framework* Ionic (Ionic-Framework, 2020). Las anteriores clases de aplicaciones permiten su desarrollo utilizando el conocimiento para el diseño web, además de la posibilidad de acceder a las funcionalidades nativas de los dispositivos, uniendo las ventajas que existen en las aplicaciones nativas y en las aplicaciones web.

En la segunda parte de la capa de presentación se utilizará el *framework* Angular (AngularFramework, 2020), para su desarrollo utilizaremos el lenguaje de programación TypeScript, el cual ofrece una estructura de código con mayor estabilidad que el uso de JavaScript puro, que, combinada con la naturaleza del *framework* basada en componentes, nos facilita la codificación de los prototipos, favoreciendo a los aspectos como la estabilidad y escalabilidad.

Capa lógica de negocios

En esta capa se definirán las clases que modelan la solución que implementamos y la lógica funcional. En general en las funcionalidades CRUD no abundan, pero serán fácilmente identificables cuando esto se aplique a clases que intervienen en los genuinos procesos funcionales del sistema. También desde aquí se emiten los mensajes hacia la capa de datos para que se concreten las operaciones de acceso a los datos almacenados y la referencia hacia el objeto que posee en su estado los datos a ser tratados.

La capa lógica de negocios hace referencia al alojamiento de los procesos que trabajarán con los datos que fluirán entre la capa de presentación y la capa de datos. Para el alojamiento de la estructura lógica, se hospedarán en un servidor con sistema operativo Linux. La estructura lógica se verá desarrollada en el ambiente de trabajo Node (Nodejs, 2020), que ofrece la posibilidad utilizar el lenguaje de programación JavaScript para el *back-end*, es aquí donde utilizando

el patrón de arquitectura de *software MVC* se desarrollan los modelos para la estructura de los datos; también, se elaboran los algoritmos con los que trabajará la información para la toma de decisiones, esto ligado a la capa de presentación elaborada en página web.

Es necesario que la capa lógica de negocios esté ligada con la capa de datos, para esto, se utilizará el *framework Express*, que servirá como *middleware* tanto para el tratamiento de los datos como para permitir el flujo de información entre la capa de presentación con la capa de datos de forma bidireccional, esto con la finalidad de que exista la interacción entre ambas capas de presentación (*Express-Framework*, 2020).

Capa de datos

La funcionalidad que debe cubrir abarca la descomposición sistemática del estado del objeto que posee los datos, a estructuras manipulables en los términos del modelo de acceso a datos utilizado y la base de datos seleccionada para persistirlos. También se encargará de gestionar las conexiones y ejecución de comandos necesarios para operar sobre la base de datos.

La capa de datos maneja todos los datos que son generados por las capas de presentación, esta capa se encarga de almacenar la información que es generada por los usuarios utilizando los modelos proporcionados por la capa lógica de negocios. El propósito de la capa de datos es mantener la disponibilidad de la información para que esta pueda ser recuperada en la capa de presentación, siendo esta tratada en la capa lógica de negocios con la finalidad de llevar solo la información que se necesita mostrar.

La existencia de la capa de datos brinda una mayor escalabilidad, ya que los procesos que tratan los datos se encuentran separados, creando distintas posibilidades para trabajar con nuevos procesos.

Con lo expuesto, se puede observar claramente la separación funcional de los aspectos relacionados con el manejo de las vistas, la lógica y los datos. No obstante, este enfoque puede ser diferente con-

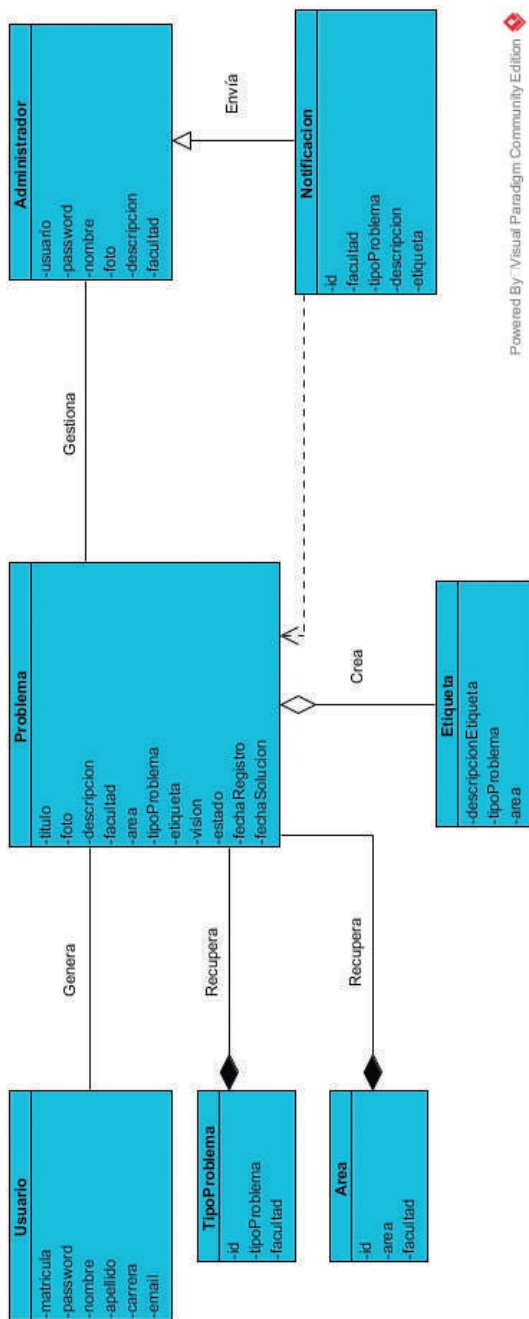
siderando las necesidades del proyecto, así como su tamaño, donde en un proyecto con mayor tamaño puede crear más capas refinadas en una nueva arquitectura con el propósito de dar más flexibilidad al diseño, así como una mejor separación de conceptos. Sin embargo, si se intentara aplicar esta clase de lógica en proyectos pequeños o medianos, lo único que se obtendría es agregar complejidad e inestabilidad al *software*.

b.2. Diagrama de clases

El diagrama de clases sirve para mostrar el comportamiento de la base de datos al conectarla a algún sistema, pues permite visualizar las distintas clases que la componen (equivalente a las tablas en la base de datos), los distintos atributos de estas, los tipos de datos que se utilizan, las relaciones entre las clases, y comúnmente las operaciones que se pueden realizar dentro de la clase (Fowler y Scott, 1999).

El formato de diagrama de clases que se utilizó (figura 3) fue basado en el Lenguaje Unificado de Modelado (UML) (Fowler y Scott, 1999), modificado para no mostrar las operaciones que se realizan dentro de esta. El diagrama de clases nos permite ver una arquitectura a nivel modelo. El diagrama de clases del prototipo refleja tanto las tareas principales que se realizarán, así como los elementos que nos permitirán clasificar la información generada por el prototipo de aplicación móvil.

Figura 3
Diagrama de clases del sistema



Powered By Visual Paradigm Community Edition

Fuente: Elaboración propia.

C) Implementación (interfaces del sistema)

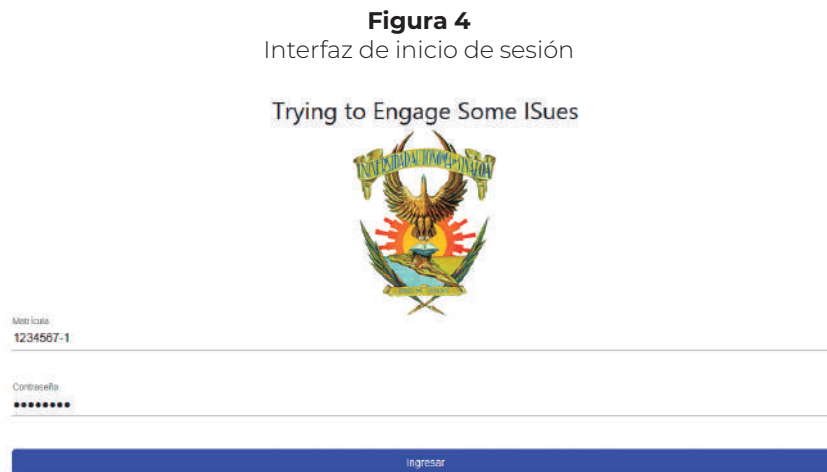
Debido a que el sistema está compuesto por dos aplicaciones, se diseñaron interfaces para la aplicación web del administrador y se diseñaron interfaces para la aplicación móvil de los usuarios.

c.1. Aplicación web (administrador)

El sitio web del sistema está conformado por la interfaz de inicio de sesión, la interfaz de inicio, la interfaz de información para la solución del problema, la interfaz de notificaciones y la interfaz de estadísticas.

Interfaz de inicio de sesión

Para iniciar sesión se requiere la matrícula (número de empleado) y la contraseña del administrador. La figura 4 presenta la interfaz de inicio de sesión.



Fuente: Elaboración propia.

Interfaz de inicio

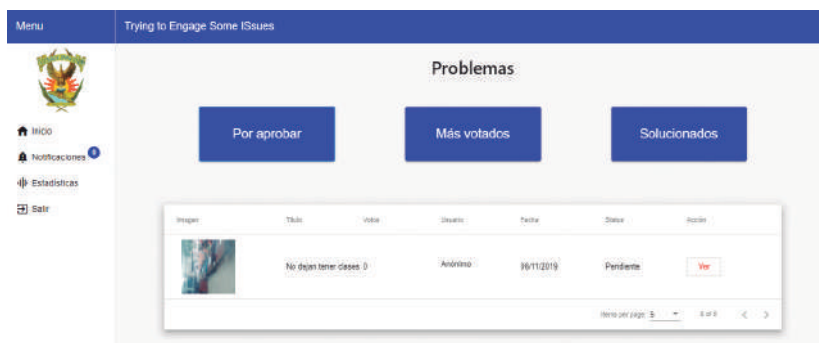
La interfaz de inicio muestra las distintas herramientas con las que el administrador podrá trabajar, aprobar problemas, solucionar problemas, observar notificaciones, ver estadísticas y el registro de todos los problemas reportados. En la figura 5 se muestra la interfaz de inicio del sistema y en la figura 6 se puede observar una interfaz con una tabla de información relacionada con los problemas que se han reportado a través de la aplicación móvil.

Figura 5
Interfaz de inicio



Fuente: Elaboración propia.

Figura 6
Interfaz con tabla de problemas



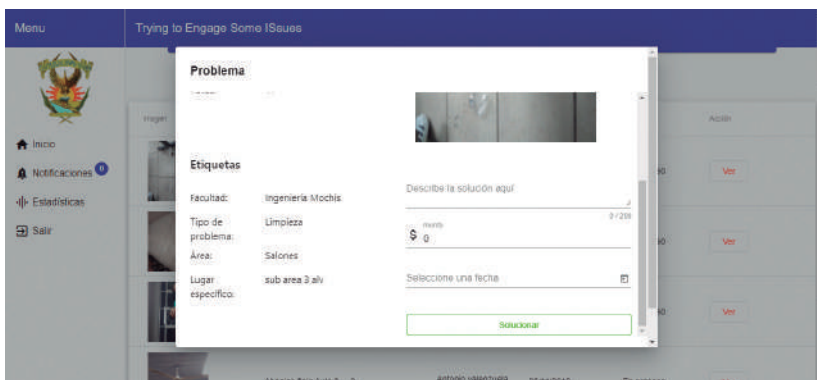
Fuente: Elaboración propia.

Interfaz de información para la solución del problema

Al momento de definir un problema como solucionado, esta interfaz permite agregar comentarios sobre la solución del problema, la fecha en la que se solucionó y la cantidad de monetaria que se gastó para llegar a la solución (esto último no es visible al público, solo funciona para la administración interna). En la figura 7 se puede observar la interfaz de información para la solución del problema.

Figura 7

Interfaz de información para la solución del problema



Fuente: Elaboración propia.

Interfaz de notificaciones

Por medio de la interfaz de notificaciones se muestran al administrador los mensajes que indican los posibles problemas futuros o relaciones en las tendencias de reportes. Esto sirve para tomar acciones preventivas. La figura 8 muestra la interfaz de notificaciones.

Figura 8
Interfaz de notificaciones



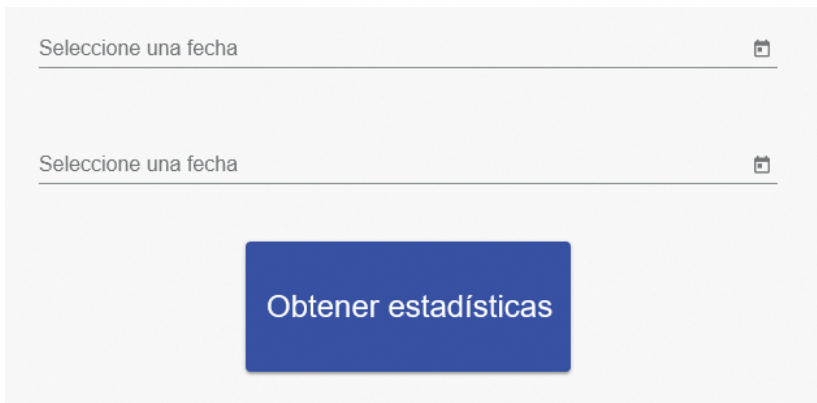
Fuente: Elaboración propia.

Estadísticas

La interfaz de estadísticas visualiza gráficamente un informe de los problemas registrados por los usuarios de la aplicación móvil. Las figuras 9 y 10 presentan la interfaz de estadísticas.

Figura 9

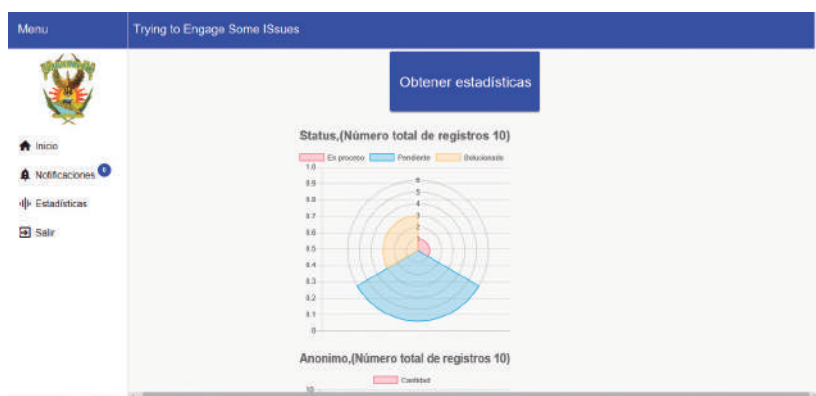
Interfaz de estadísticas (informe de estadísticas por rango de fechas)



The image shows a mobile application interface for generating statistics. It consists of two identical date selection fields, each with the placeholder text "Seleccione una fecha" and a calendar icon on the right. Below these fields is a prominent blue button with the white text "Obtener estadísticas".

Fuente: Elaboración propia.

Figura 10
Interfaz de estadísticas



Fuente: Elaboración propia.

c.2. Aplicación móvil (estudiantes, docentes)

La aplicación móvil está conformada por la interfaz de registro, la interfaz de inicio de sesión, la interfaz de inicio, la interfaz de menú y la interfaz para reportar problemas.

Interfaz de registro

Para realizar el registro en la aplicación se debe contar con una matrícula (número de cuenta de estudiantes y número de empleo de docentes) vigente de la universidad, esto con la finalidad de mantener los reportes limitados a estudiantes y docentes de la institución. La figura 11 presenta la interfaz de registro de la aplicación móvil.

Figura 11
Interfaz de registro

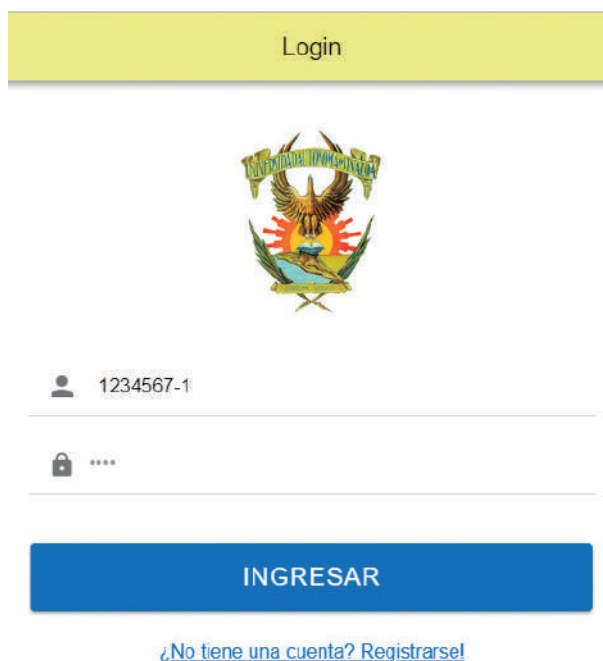
The image shows a mobile application registration screen. At the top, there is a yellow header bar with a back arrow on the left and the title 'Registro' in the center. Below the header, there are seven input fields, each with a small icon to its left: a person icon for 'Matrícula', an envelope icon for 'E-mail(para recuperación de contraseña)', a lock icon for 'Contraseña', another lock icon for 'Repetir contraseña', a pencil icon for 'Nombre(s)', another pencil icon for 'Apellido(s)', a graduation cap icon for 'Carrera', and a person icon for 'Tipo de usuario'. A red error message '*Las contraseñas no coinciden' is displayed to the right of the password fields. A vertical scrollbar is visible on the right side of the form.

Fuente: Elaboración propia.

Interfaz de inicio de sesión

La interfaz de inicio de sesión permite ingresar al usuario a la aplicación móvil por medio de su matrícula y clave de acceso. En la figura 12 se puede observar la interfaz de inicio de sesión.

Figura 12
Interfaz de inicio de sesión

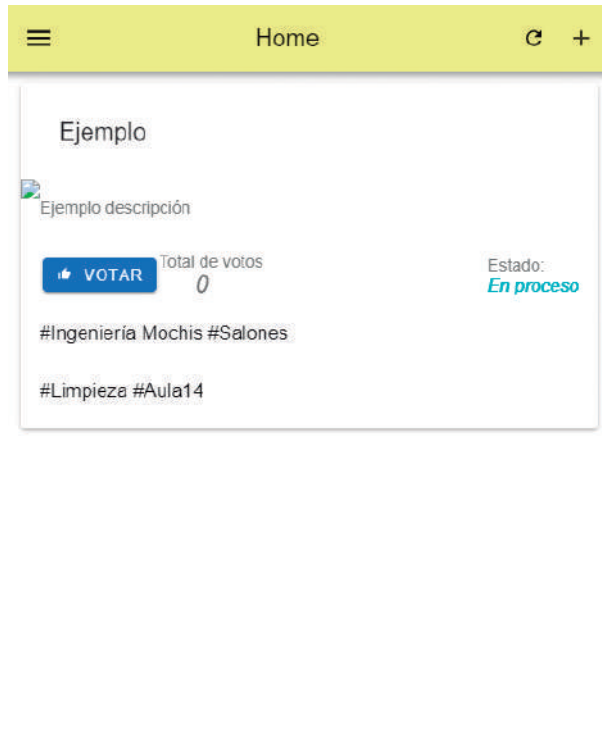


Fuente: Elaboración propia.

Interfaz de inicio

La interfaz de inicio muestra los problemas reportados en la facultad a la que pertenece el usuario. Además, el usuario puede votar por los problemas que considere relevantes o sean de su interés. La figura 13 muestra la interfaz de inicio de la aplicación móvil.

Figura 13
Interfaz de inicio

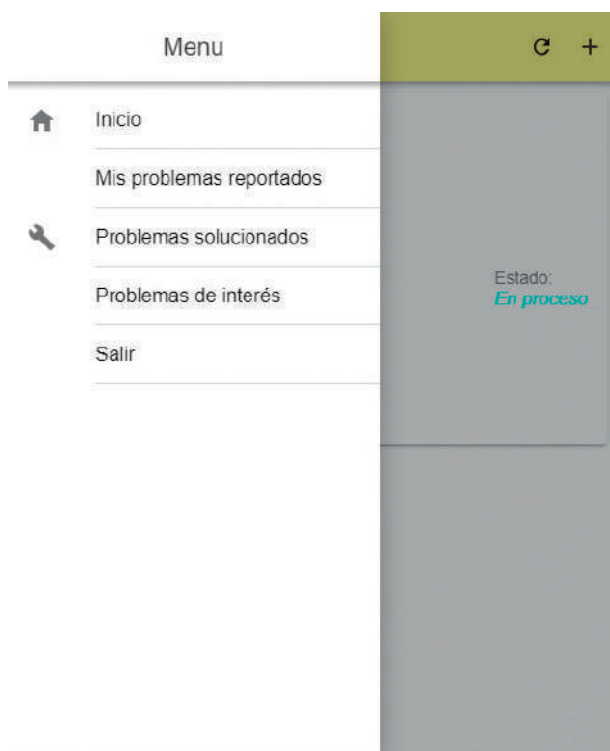


Fuente: Elaboración propia.

Interfaz de menú

La interfaz de menú del usuario le permite navegar entre los problemas que se han reportado, los problemas que se han solucionado y los problemas por los que ha votado. En la figura 14 se puede observar la interfaz de menú desde la aplicación móvil.

Figura 14
Interfaz de menú



Fuente: Elaboración propia.

Interfaz para reportar problemas

La interfaz para reportar problemas permite al usuario crear un nuevo registro de problema. El usuario debe llenar los campos definidos (título del problema, imagen, descripción del problema, facultad, área, tipo de problema, etiqueta y usuario) y tomar una foto para una mejor referencia. La figura 15 presenta la interfaz para reportar problemas de la aplicación móvil.

Figura 15
Interfaz para reportar problemas

The image shows a mobile application interface for reporting a problem. The title bar is yellow and contains the text 'Reportar problema' and a close button (X). Below the title bar, there are several input fields and a toggle switch:

- Título del problema:** A text input field with a pencil icon on the left and a camera icon on the right.
- Describe el problema aquí:** A text area with a speech bubble icon on the left.
- Facultad:** A dropdown menu with a graduation cap icon on the left.
- Area:** A dropdown menu with a document icon on the left.
- Tipo de problema:** A dropdown menu with a wrench icon on the left.
- Etiqueta personalizada:** A text input field with a tag icon on the left.
- Anónimo:** A toggle switch with a leaf icon on the left.

At the bottom of the form is a blue button with a white document icon and the text 'REPORTAR PROBLEMA'.

Fuente: Elaboración propia.

D) Pruebas (evaluación)

Para la evaluación del sistema se aplicaron algunas preguntas del inventario de medición de usabilidad de *software* (SUMI, por sus siglas en inglés: *Software Usability Measurement Inventory*). SUMI es una solución al problema recurrente de medir la percepción de los usuarios sobre la usabilidad del *software*. Proporciona un método válido y confiable para la comparación de productos de la competencia y diferentes versiones del mismo producto, así como información de diagnóstico para desarrollos futuros. Está destinado a ser adminis-

trado a una muestra de usuarios que han tenido alguna experiencia en el uso del *software* a evaluar (Kirabowski y Corbett, 1993).

Técnica de muestreo

Para la aplicación de la prueba se decidió utilizar una técnica no probabilística por conveniencia. La técnica de muestreo no probabilístico por conveniencia permite seleccionar aquellos casos accesibles que acepten ser incluidos. Esto, fundamentado en la conveniente accesibilidad y proximidad de los sujetos para el investigador (Hernández Sampieri, Fernández Collado y Baptista Lucio, 2014).

Participantes y muestra

Como sujetos para la prueba se seleccionaron dos grupos de primer año de la Facultad de Ingeniería Mochis (FIM) de la Universidad Autónoma de Sinaloa (UAS) que estudian la carrera de licenciatura en Ingeniería de Software (LIS), siendo un total de 43 estudiantes entre ambos grupos. De estos, solamente 28 pudieron registrarse debido a algunos problemas de compatibilidad con algunos dispositivos, de este grupo dos estudiantes decidieron no participar en el experimento, quedando un total de 26 estudiantes. Dado que los participantes eran de primer año, los conocimientos relacionados en cuanto al desarrollo de *software* eran básicos, sin embargo, contaban con la experiencia suficiente para el uso de la aplicación móvil.

Metodología

Para los estudiantes se prepararon algunos recursos visuales de capacitación que ayudarían a dar una explicación inicial de la aplicación móvil. Se planteó cuál es el objetivo y con qué componentes se cuenta para alcanzarlo. Para una primera prueba, se les proporcionó Internet y un código QR con el cual se podría descargar la aplicación móvil.

Mientras los participantes realizaban la instalación, se les dio un breve tutorial de cómo realizar el registro de usuario en la aplicación.

Cuando los estudiantes hicieron su registro e iniciaron sesión, se les dio la oportunidad de que navegaran entre las opciones para saber si existían dudas sobre sus componentes y su función. Se les pidió que, como prueba, registraran algunos problemas que detectaran dentro de sus aulas y algunos al azar, esto ayudaría a explicar de mejor manera la parte de gestión de problemas registrados a los administradores del sistema.

Mientras los estudiantes realizaban la tarea, se conversó con ellos sobre el funcionamiento de algunos componentes, por ejemplo: “¿Por qué al momento de registrar un problema no es posible subir una foto desde la galería del dispositivo?”. la razón es porque se quería estar seguros de que, al momento de realizar el registro de un problema, la persona esté presente en dicho lugar y la foto represente realmente el conflicto que se quiere exponer.

Por último, cuando los participantes generaron algunos problemas, se habilitaron algunos de estos para que ellos pudieran interactuar con los problemas de otros usuarios por medio de las votaciones. Al finalizar las tareas de capacitación, se les agradeció por su cooperación y se indicó que la aplicación estaría disponible durante tres semanas para la recolección de información. Esto con la intención de conocer la experiencia de usuario, qué tan útil es en la facultad, recabar opiniones sobre el funcionamiento y de los datos que maneja.

De manera paralela, también se expuso al administrador cómo los problemas registrados por los estudiantes iban a ser gestionados desde la aplicación web durante las tres semanas de prueba. Esta sin duda es una de las partes más importantes del sistema, ya que con el apoyo del algoritmo Apriori se detectaron problemas que requerían la atención del administrador de la FIM-UAS para la toma de decisiones.

Finalmente, al concluir las tres semanas de pruebas se aplicó la encuesta SUMI a los 26 participantes que utilizaron la aplicación móvil.

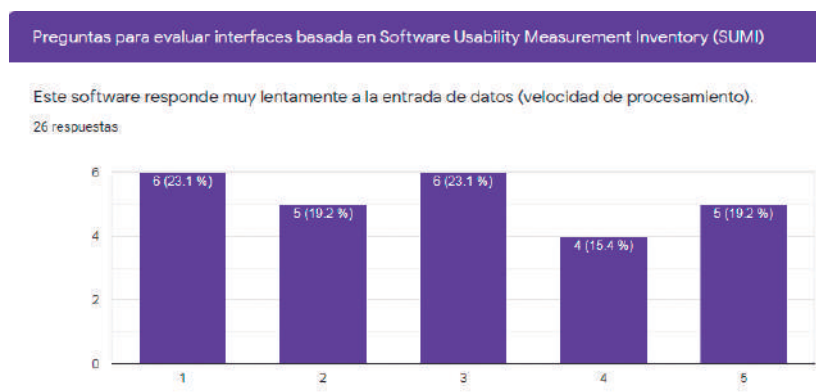
Resultados

Al concluir las tres semanas de experimentación se aplicaron ocho preguntas del test de usabilidad SUMI para conocer el punto de vista de los usuarios con respecto a la calidad de uso de la aplicación móvil: 1. velocidad de procesamiento, 2. ayuda, 3. confiabilidad, 4. claridad de la información, 5. organización de menús, 6. diseño de interfaces, 7. accesibilidad, y 8. mensajes de error. Es importante mencionar que se modificaron las opciones de respuesta. Así, las opciones de respuesta de cada pregunta van del 1 al 5, donde 1 significa “totalmente en desacuerdo” y 5 “totalmente de acuerdo”.

La figura 16 presenta los resultados de la primera pregunta: “¿Este *software* responde muy lentamente a la entrada de datos?”. Como se puede apreciar, el 34.6 % de los usuarios (concentrado de respuestas 4 y 5) consideran que existe una lenta velocidad de procesamiento.

Figura 16

Pregunta 1. Velocidad de procesamiento



Fuente: Elaboración propia.

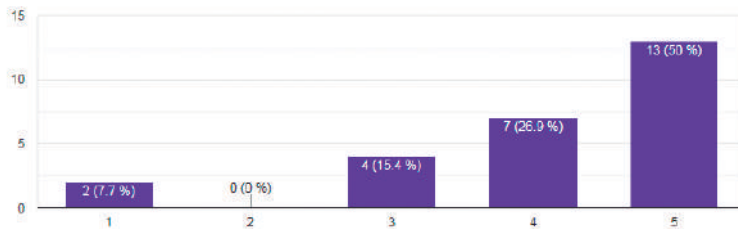
La figura 17 presenta los resultados de la segunda pregunta: “¿Las instrucciones y ayudas son útiles?”. Como se puede apreciar, los

resultados relacionados con la ayuda muestran que el 76.9 % de los usuarios encontraron útiles las indicaciones que se le ofrecen.

Figura 17
Pregunta 2. Ayuda

Las instrucciones y ayudas son útiles (ayuda).

26 respuestas



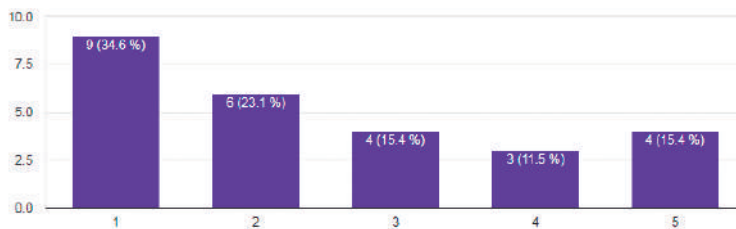
Fuente: Elaboración propia.

La figura 18 presenta los resultados de la tercera pregunta: “¿El software se ha parado alguna vez de forma inesperada?”. Como se puede ver, los resultados relacionados con la confiabilidad muestran que solo el 26.9 % tuvieron algún problema con la aplicación.

Figura 18
Pregunta 3. Confiabilidad

El software se ha parado alguna vez de forma inesperada (confiabilidad).

26 respuestas



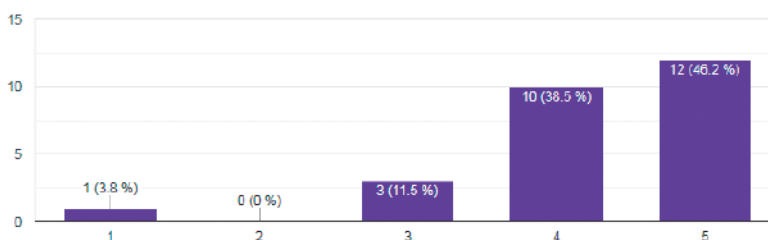
Fuente: Elaboración propia.

La figura 19 presenta los resultados de la cuarta pregunta: “¿La forma en la que el sistema presenta la información es clara y comprensible?”. En la figura se pueden observar los resultados relacionados con la claridad de la información (a la que los usuarios pueden acceder), donde se muestra que el 84.6 % de los usuarios pudieron entender fácilmente los datos mostrados por la aplicación.

Figura 19
Pregunta 4. Claridad de la información

La forma en la que el sistema presenta la información es clara y comprensible (claridad).

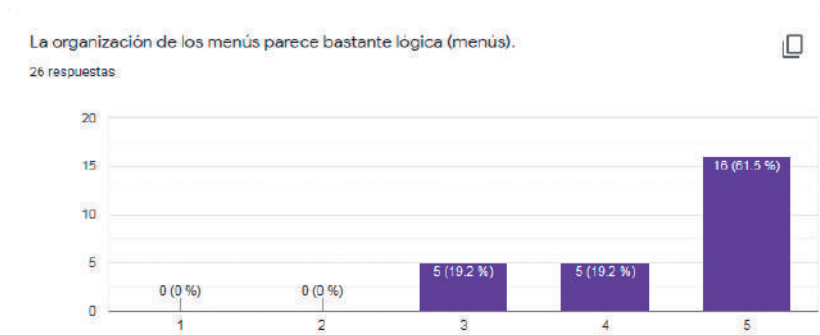
26 respuestas



Fuente: Elaboración propia.

La figura 20 presenta los resultados de la quinta pregunta: “¿La organización de los menús parece bastante lógica?”. Los resultados arrojan que el 80.8 % de los usuarios consideraron la organización de menús de navegación bastante lógica.

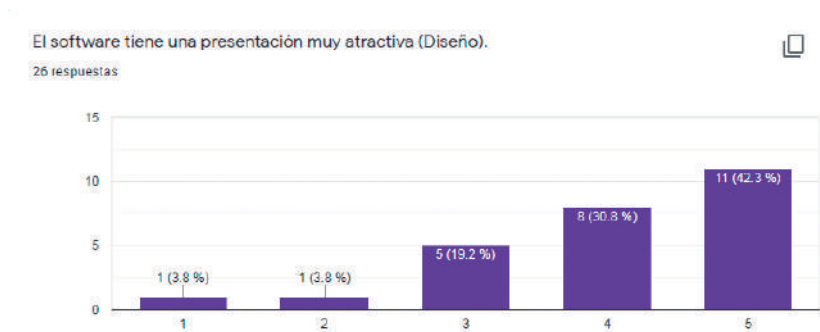
Figura 20
Pregunta 5. Organización de menús



Fuente: Elaboración propia.

La figura 21 presenta los resultados de la sexta pregunta: “¿El *software* tiene una presentación muy atractiva?”. Los resultados arrojan que el 73.1 % de los usuarios consideraron el diseño de las interfaces muy atractivas.

Figura 21
Pregunta 6. Diseño de interfaces

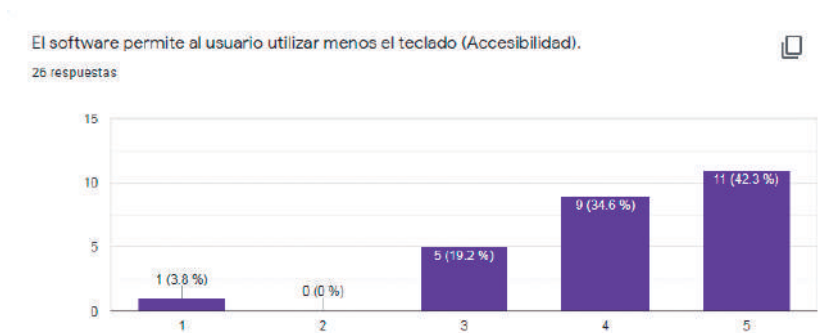


Fuente: Elaboración propia.

La figura 22 presenta los resultados de la séptima pregunta: “¿El *software* permite al usuario utilizar menos el teclado?”. El 76.9 % de los

usuarios tuvieron una respuesta positiva en cuanto a la característica de accesibilidad.

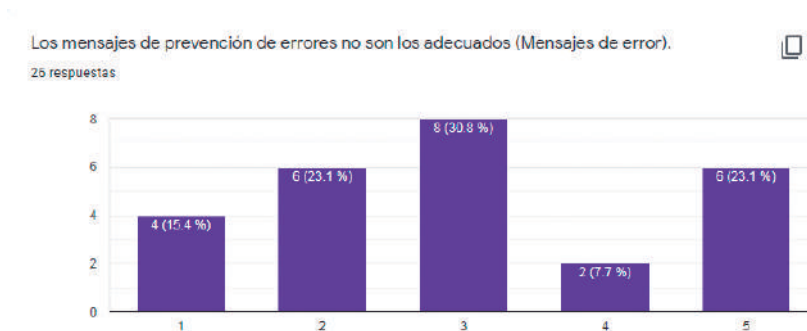
Figura 22
Pregunta 7. Accesibilidad



Fuente: Elaboración propia.

La figura 23 presenta los resultados de la octava pregunta: “¿Los mensajes de prevención de errores no son los adecuados?”. Los resultados indican variadas opiniones relacionadas con los mensajes de error, donde la mayoría se concentra en el punto neutro.

Figura 23
Pregunta 8. Mensajes de error

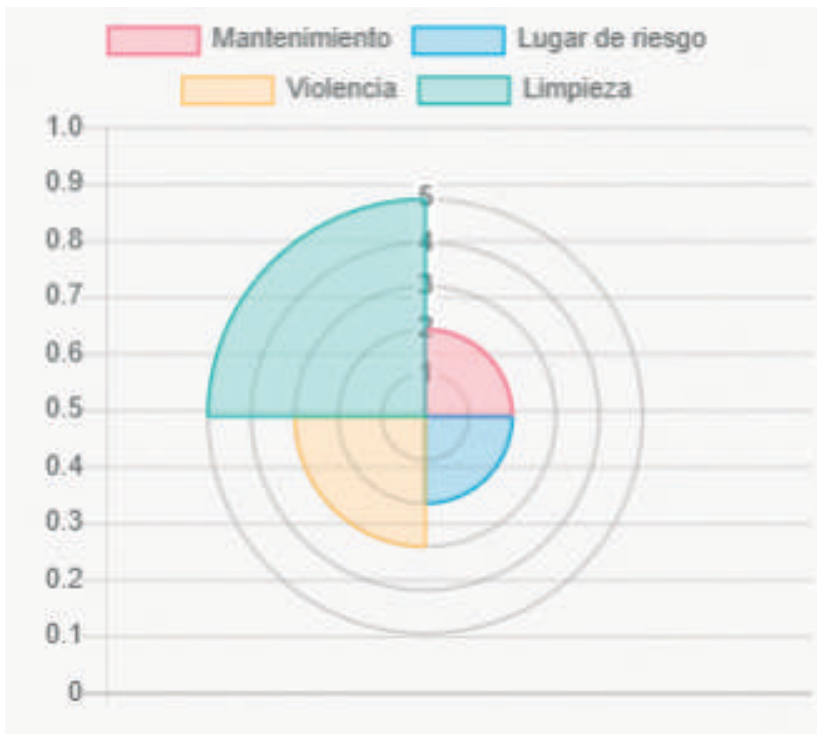


Fuente: Elaboración propia.

En la parte del administrador (sitio web de gestión) se controló el flujo de todos los problemas que se iban registrando y se identificó que solo el 5 % de los problemas registrados no eran aptos para ser habilitados, esto muestra que los usuarios hicieron un uso debido de la aplicación.

En la figura 24 se observan las estadísticas de los datos registrados, donde se aporta al administrador los tipos de problemas más recurrentes. Por último, se hizo un contraste de la información existente en las tablas con las notificaciones generadas por el algoritmo Apriori que dieron sugerencias precautorias para los administradores.

Figura 24
Estadísticas generadas por la página web



Fuente: Elaboración propia.

Discusión y conclusiones

Discusiones

Las tres semanas que duró la prueba permitió conocer la percepción de las personas sobre la calidad de uso de la aplicación móvil. Todos los participantes reportaron problemas relacionados con instalaciones sucias, fallas de abanicos, problemas con equipos de cómputo, entre otros.

En este contexto, la aplicación tuvo un correcto funcionamiento, creando información clara y concisa que permitió al algoritmo Apriori identificar problemas recurrentes para la toma de decisiones de una forma eficiente para crear medidas de precaución por parte del administrador. Al mismo tiempo, indica que tanto la parte del *front-end* (la aplicación móvil) como el *back-end* (el servidor y base de datos) funcionan correctamente y tienen una estabilidad aceptable.

Las dificultades identificadas por el 34.6 % de los usuarios (pregunta 1) en la velocidad de procesamiento de los datos pudieron deberse sobre todo a factores externos como la velocidad de Internet. No obstante, el 42.3 % de los usuarios la considera una velocidad de procesamiento de datos aceptable, y al 23.1 % le resulta indiferente.

Con relación en las instrucciones que aporta la aplicación móvil, donde el 76.9 % de los usuarios (pregunta 2) encontraron indicaciones útiles, esto puede deberse a la experiencia de los usuarios en el uso de aplicaciones móviles y a las habilidades de los usuarios para encontrar y resolver la duda que tenían. Sin embargo, el 7.7 % de los usuarios no encontraron de utilidad las indicaciones, y al 15.4 % le resulta indiferente.

Los problemas de confiabilidad que tuvieron el 26.9 % de los usuarios (pregunta 3) durante la prueba puede deberse a la saturación de memoria por diversas aplicaciones ejecutándose o el retardo en la recuperación de datos del servidor, ocasionando con ello una

falla en la aplicación. En contraparte, el 57.7 % de los usuarios considera que el *software* funciona bien, y al 15.4 % le resulta indiferente.

Por otra parte, un 84.6 % de los usuarios (pregunta 4) pudieron entender fácilmente los datos mostrados por la aplicación móvil, esto es debido al adecuado diseño empleado para la presentación de los datos. No obstante, el 3.8 % de los usuarios considera que la información no es clara y comprensibles, y al 11.5 % le resulta indiferente.

En la organización de los menús de navegación, el 80 % de los usuarios (pregunta 5) los consideraron lógicos, esto se debe a que desde que se inició con el desarrollo del sistema se consideraron las recomendaciones de usabilidad en el diseño de aplicaciones móviles. Solamente al 19.2 % de los usuarios le resulta indiferente.

Por otro lado, en el diseño de las interfaces, el 73.1 % de los usuarios (pregunta 6) las consideraron atractivas, esto se debe a que siempre se cuidó que el diseño del sistema siguiera las recomendaciones de usabilidad en el diseño de aplicaciones móviles. En contraparte, el 7.7 % de los usuarios considera que las interfaces no son atractivas, y para al 19.2 % le resulta indiferente.

Con relación en la accesibilidad, el 76.9 % de los usuarios (pregunta 7) considera que la aplicación móvil evita el uso del teclado. Solamente el 3.8 % de los usuarios considera que requiere el uso del teclado, y al 19.2 % le resulta indiferente.

Finalmente, en el manejo de errores de la aplicación, el 30.8 % de los usuarios (pregunta 8) consideraron que los mensajes de error no fueron adecuados. Sin embargo, el 38.5 % de los usuarios consideraron que los mensajes de error fueron bien manejados, y al 30.8 % le resulta indiferente.

Conclusiones

En este trabajo se desarrollaron dos sistemas que trabajan en conjunto para facilitar la retroalimentación de problemas en una comunidad universitaria:

1. Una aplicación móvil que permite a los estudiantes reportar problemas administrativos de la facultad, ya sean problemas de limpieza, lugares de riesgo, infraestructura, violencia, mantenimiento, entre otros. La aplicación permite a los estudiantes interactuar con los reportes realizados por otros usuarios, pudiendo ver si un problema que planeaban reportar ya se encuentra registrado en la aplicación, y de ser así, poder otorgar su voto, dando a entender que ese problema es de interés para otros.
2. Un sistema de soporte a las decisiones que opera a través de una aplicación web y que gestiona los reportes generados por los estudiantes. El sistema primero deja los reportes en un estado de pendiente, el cual bloquea la visibilidad de este dentro de la aplicación para filtrar aquellos problemas que no tengan la intención de representar un hecho real, sean reportes que ya se han hecho anteriormente o se encuentren solucionados. Además, el sistema genera notificaciones al administrador sobre patrones de problemas que podrían suceder a partir de la información analizada por el algoritmo Apriori. Por último, el sistema crea estadísticas que facilitan la lectura de los datos.

La combinación de ambos sistemas mostró como resultado que la comunicación entre los estudiantes y los administradores es más eficiente. En la perspectiva de la administración, la prioridad de los problemas hace más fácil saber qué inconvenientes deben tratarse primero. Por otra parte, los estudiantes y la administración que conforman la facultad tuvieron una sensación de cercanía con la institución para mejorar el ambiente y las instalaciones.

Este sistema fue diseñado pensando que puede aplicarse en distintas facultades de una ciudad universitaria. Sin embargo, también puede ser aplicado en otras instituciones que cumplan con los requerimientos del sistema.

El funcionamiento de la aplicación móvil y la aplicación web de gestión puede ser optimizado en un futuro en aspectos como: 1. el procesamiento de las imágenes, disminuyendo su peso, redu-

ciendo el espacio que ocupan y el tiempo que necesita el sistema para mostrarlas; 2. los valores utilizados en el algoritmo Apriori pueden ser mejorados, modificando sus valores de soporte y confianza para obtener resultados más precisos con la información que se tiene.

Referencias

- Angular (2020), “Angular”, *Angular*, 1 de agosto. Disponible en: <https://angular.io/>
- AngularFramework (2020), “One Framework Mobile & Desktop”, *One framework mobile & desktop*, 1 de agosto. Disponible en: <https://angular.io/>
- Balakrishna, C. (2012), “Enabling Technologies for Smart City Services and Applications”, *Sixth International Conference on Next Generation Mobile Applications, Services and Technologies*, pp. 223-226, Cardiff.
- Bayardo, R. (1998), “Efficiently Mining Long Patterns from Databases”, *ACM-SIGMOD Int’l Conf. on Management of Data*, pp. 85-93. Consultado: septiembre 18, 2019.
- Bootstrap (2020), “Build Fast, Responsive Sites with Bootstrap”, *Build Fast, Responsive Sites with Bootstrap*, 1 de agosto. Disponible en: <https://getbootstrap.com/>
- Brooke, J. (1996), “SUS-A Quick and Dirty Usability Scale”, *Usability Evaluation in Industry*, 189(194), pp. 4-7.
- Chichernea, V. (2014), “The Use of Decision Support Systems (DSS) in Smart City Planning and Management”, *Romanian Economic Business Review, Romanian-American University*, pp. 238-251.
- Desouza, K. y A. Bhagwatwar (2012), “Citizen Apps to Solve Complex Urban Problems”, *Journal of Urban Technology*.
- Doria, M. V.; Montejano, G. A. y C. I. Inchaurredo (2014), “Modelo de repositorio abierto en Ingeniería de Software para el ámbito universitario”, *XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO) - I Simposio Argentino de Tecnología y Sociedad (STS)*, pp. 125-139, Argentina, Buenos Aires.
- Escobar Acevedo, A. (2009), *Métodos de refinamiento de la clasificación translingüe de documentos (Tesis de maestría)*, septiembre, Puebla, Tonantzintla, Instituto Nacional de Astrofísica, Óptica y Electrónica INAOE.

- ExpressFramework (2020), “Express infraestructura web rápida, minimalista y flexible para Node.js”, *Express infraestructura web rápida, minimalista y flexible para Node.js*, 1 de agosto. Disponible en: <https://expressjs.com/es/>
- Expressjs (2020), “Express”, *Express*, 1 de agosto. Disponible en: <https://expressjs.com/es/>
- Fan, H. y Y. Qin (2018), *Research on Text Classification Based on Improved TF-IDF Algorithm*, pp. 501-506, Atlantis Press, DOI:<https://doi.org/10.2991/ncce-18.2018.79>
- Fowler, M. y K. Scott (1999), *UML gota a gota*, Alhambra Mexicana.
- GmbH, d. (2020), “DSpace”, *DSpace*. Disponible en: <https://www.dspace.com/en/pub/home.cfm>. Consultado: enero 6, 2020.
- Hernández Sampieri, R.; Fernández Collado, C. y M. Baptista Lucio (2014), *Metodología de la Investigación*, 6ª ed., McGraw-Hill.
- IonicFramework (2020), “Ionic Community”, *Ionic Community*, 1 de agosto. Disponible en: <https://ionicframework.com/community>
- Jisc (2020), “Jisc”. Disponible en: http://http://v2.sherpa.ac.uk/view/repository_by_country/Mexico.html. Consultado: febrero 4, 2020.
- Kirabowski, J. y M. Corbett (1993), “SUMI: The Software Usability Measurement Inventory”, *British Journal of Educational Trchnology*, 24(3), pp. 210-212.
- Lemos, A. y N. Veras (2018), “Cidadão Sensor e Cidade Inteligente: Análise dos Aplicativos Móveis da Bahia”, *Revista Famecos*.
- Lucid, S. (2020), “Lucidchart”. Disponible en: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>. Consultado: febrero 12, 2020.
- Maciá Pérez, F. (2017), *Smart University - Hacia una universidad más abierta*, Alfaomega, Marcombo.
- Meaning Cloud (2020), “Meaning Cloud”, Meaning Cloud, 1 de agosto. Disponible en: <https://www.meaningcloud.com/es>
- Medina González, A. (2017), “Implementación de un repositorio digital para el entorno local de la Facultad de Comunicación de la Universidad de la Habana”, 13(2), pp. 202-214.
- MongoDB (2020), “MongoDB”, *MongoDB*, 1 de agosto. Disponible en: <https://www.mongodb.com/es>
- N. Ocholla, D. (2011), “An Overview of Issues, Challenges and Opportunities of Scholarly Publishing in Information Studies in Africa”, *African Journal of Library, Archives & Information Science*, 21(1), pp. 1-16.

- Nodejs (2020), “Node.js”, *Node.js*, 1 de agosto. Disponible en: <https://nodejs.org/es/>
- (2020), “Node.js® es un entorno de ejecución para JavaScript”, *Node.js® es un entorno de ejecución para JavaScript*, 1 de agosto. Disponible en: <https://nodejs.org/es/>
- Parra-Valeancia, J.; Guerrero, C. y D. Rico (2017), “IoT: Una aproximación desde ciudad inteligente a universidad inteligente”, *Revista Ingenio UFPSO*, 13, pp. 9-15.
- Pérez Abelleira, M. A. y A. C. Cardoso (2011), “Categorización automática de documentos”, *Simposio Argentino de Inteligencia Artificial*, pp. 20-31, Argentina, Salta.
- Pressman, R. (2010), *Ingeniería de software. Un enfoque práctico*, 7ª ed., México D.F., Editorial McGraw-Hill.
- Sommerville, I. (2005), *Ingeniería del software*, 7ª ed., Madrid, Pearson.
- Southampton, U. O. (2020), “eprints.com”. Disponible en: <https://www.eprints.org/uk/>. Consultado: marzo 20, 2020.
- Texier, J.; De Giusti, M. R.; Oviedo, N. F.; Villarreal, G. L. y A. J. Lira (2002), “El uso de repositorios y su importancia para la educación en Ingeniería”, *World Engineering Education Forum (WEEF)*, Argentina, Buenos Aires.
- Trstenjak; Bruno, M.; Sasa, D. Dzenana (2014), “KNN with TF-IDF Based Framework for Text Categorization”, *24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013*, pp. 1356-1364, Eslovenia, Elsevier, DOI: 10.1016/j.proeng.2014.03.129
- Yolis, E.; Britos, P. y Perichisky (2008), “Algoritmos genéticos aplicados a la categorización automática de documentos”, *Revista Eletrônica de Sistemas de Informação*, 2(2), pp. 1-15, DOI: 10.21529/RESI.2003.0202001

4

REPOSITORIO DIGITAL Y CLASIFICACIÓN AUTOMÁTICA DE TESIS*

YOBANI MARTÍNEZ RAMÍREZ¹

ALAN RAMÍREZ NORIEGA²

ANA MARÍA HERNÁNDEZ RUIZ³

MARÍA DEL CARMEN RODRÍGUEZ APODACA⁴

Resumen

Un repositorio es un sitio en donde se pueden almacenar y organizar distintos tipos de documentos y datos con información en su totalidad digital, dando paso a lo que conocemos como repositorio digital. En la actualidad, los repositorios son utilizados frecuentemente

* Derivado del proyecto de investigación: "Actividades de Investigación del Cuerpo Académico uas-ca-295".

1 Universidad Autónoma de Sinaloa, correo electrónico: yobani@uas.edu.mx

2 Universidad Autónoma de Sinaloa, correo electrónico: alandramireznoriega@uas.edu.mx

3 Universidad Autónoma de Sinaloa, correo electrónico: amhr_1997@hotmail.com

4 Universidad Autónoma de Sinaloa, correo electrónico: mdcra1997@outlook.com

con fines escolares y académicos, utilizados por alumnos, maestros, doctores, y otro universo de personas con la decisión de compartir su conocimiento. En este sentido, con el objetivo de hacer más ágil el proceso de archivado de los repositorios, este proyecto propone la categorización automatizada de un repositorio digital por medio del desarrollo de una herramienta de *software*. Mediante la tecnología propuesta, es posible decidir sobre los pesos a los términos y palabras que asignará un administrador asegurando que el documento sea categorizado de forma correcta. La plataforma de repositorio de tesis digital es una plataforma web diseñada para el uso público de la comunidad universitaria. Se integraron aspectos de interés de los alumnos, tales como una búsqueda de documentos amigable e intuitiva, descarga de documentos sencilla y rápida, y categorización automática con o sin conexión a Internet. Se probó la herramienta con tesis de distintas temáticas realizando pruebas con usuarios sin experiencia en la aplicación. Se obtuvieron resultados positivos en pruebas de usabilidad y funcionalidad. Por lo tanto, implementar una plataforma como esta en escuelas sería un paso más para compartir conocimiento.

Palabras clave: repositorio digital, categorización automatizada, tesis, meaning cloud.

Abstract

A repository is a place where different types of documents and data can be stored and organized. When this information is digital, it is called a digital repository. Today, repositories are frequently used for school and academic purposes, used by students, teachers, doctors, and others with the decision to share their knowledge. In this sense, with the aim of streamlining the repository organization process, this project proposes the automatic categorization of a digital repository through the development of a software tool. Using the proposed technology, it is possible to decide on the weights of the terms and

words assigned by an administrator, ensuring that the document was correctly categorized. The digital thesis repository platform is a web platform designed for public use by the university community. Aspects of student interest were integrated, such as a friendly and intuitive document search, simple and fast document download, and automatic categorization with or without an Internet connection. The tool was tested with theses on different subjects, carrying out tests with users without experience in the application. Positive results were obtained in usability and functionality tests. Therefore, implementing a platform like this in schools would be one more step to share knowledge.

Keywords: digital repository, automation categorization, thesis, meaning cloud.

Introducción

El desarrollo de distintos medios de expansión del conocimiento ha provocado un aumento colosal de textos electrónicos e impresos, dichas publicaciones conforman repositorios desmedidos de información en continuo crecimiento. Los repositorios son depósitos de archivos digitales de diferentes tipologías para accederlos, difundirlos y preservarlos (Texier, De Giusti, Oviedo, Villarreal y Lira, 2002).

Los repositorios digitales institucionales tienen como objetivo poner la producción científica de una institución a disposición de quienes la deseen consultar en su comunidad. Con ello se disminuyen las barreras en torno al acceso y se preserva a largo plazo la producción científica, como parte de la memoria investigativa de la institución a modo de registro de sus resultados alcanzados a lo largo del tiempo (Medina González, 2017).

De acuerdo con Ocholla (2011), la misión de la mayoría de los repositorios digitales es establecer un mejor acceso, capacidad de búsqueda, usabilidad, y visibilidad de sus resultados de investigación para aquellos con acceso a Internet. Con el objetivo de aprovechar

la información incluida en los documentos (del repositorio digital), se crean distintas tendencias de investigación, tales como (Escobar Acevedo, 2009):

1. Los sistemas de recuperación de información para consultar datos relevantes en enormes repositorios de documentos.
2. Búsqueda de respuestas donde se permite a los usuarios plantear preguntas en lenguaje natural.
3. Generación automática de resúmenes, que busca extraer líneas importantes de los documentos a fin de apresurar su lectura.
4. La clasificación de textos, que se encarga de etiquetar a los textos de forma automática. La clasificación automática de textos en el llamado aprendizaje computacional (o aprendizaje automático); es comúnmente una tarea supervisada, lo que significa que no solo se conocen las categorías, sino que debe contarse con un conjunto de entrenamiento.

En este último punto, nuevos desafíos se han descubierto debido a la necesidad de clasificar automáticamente textos en repositorios digitales, los cuales se basan en opiniones y sentimientos de las personas, y no un tanto relacionados con la temática que corresponde. Este tipo de clasificación es conocida como *sentiment classification*, *sentiment analysis* y *opinion mining*, y ha ocasionado recientes investigaciones a causa del gran interés por parte de los investigadores del procesamiento del lenguaje natural.

Por otra parte, actualmente en México se cuenta con 43 repositorios digitales alrededor del país, de los cuales 36 son repositorios institucionales, dos disciplinarios, y el resto son conocidas como de agregado; todas hacen uso de distintas plataformas las cuales ya están prediseñadas para instituciones que quieran implementar un repositorio. Las plataformas que utilizan en su mayoría son DSpace (GmbH, 2020) como la principal, EPrints (Southampton, 2020), y el resto son desarrolladas de forma independiente. Al ser plataformas ya prediseñadas, no cuentan con ninguna “inteligencia” que les ayude a automatizar sus búsquedas o categorizaciones de sus artícu-

los, mientras que los repositorios que son desarrollados de manera independiente son desarrollados con alguna característica que facilite la búsqueda de archivos y la organización de estos (Jisc, 2020).

En este orden de ideas, ante la importancia de los repositorios digitales y con la intención de preservar las tesis digitales de licenciatura y posgrado de una institución educativa, y con ello hacer accesible los resultados de investigación, en el presente trabajo se emprende el proyecto de desarrollo e implementación de un sistema de repositorio digital de tesis con categorización automática, denominado sistema RT. Así, el proyecto pretendió responder las siguientes preguntas: ¿cuál es la metodología más adecuada en la construcción de un sistema de repositorio digital de tesis con categorización automática?, ¿qué herramientas de desarrollo de *software* libre (o de bajo costo) pueden emplearse?, ¿qué puntuación de usabilidad del sistema es la más aceptable para que el sistema sea utilizado?

Por lo anterior, se consideró necesario el desarrollo de un sistema sobre plataforma web en el cual puedan almacenarse las investigaciones de tesis y categorizarlas automáticamente para afinar las consultas requeridas. Así también, hacer admisible la incorporación de imágenes, vídeos e incluso la participación de otros usuarios de una manera segura y confiable. Adicionalmente, se buscó que las consultas a este sistema podrán seguir operando a nivel institucional incluso con la ausencia de conexión a Internet.

Revisión de la literatura

A continuación se abordan las aportaciones científicas más relevantes que influyen en este trabajo de investigación:

Según los autores Fan y Qin (2018), para el algoritmo TF-IDF (Término de Frecuencia – Frecuencia Inversa del Documento), académicos han realizado diversos trabajos de mejora. El resultado principal es la mejora del algoritmo basado en la distribución de palabras clave dentro de la clase y varias clases. Muchos estudiosos

se centran en la mejora de las IDF. Se propone la separación normal para reemplazar la parte IDF del algoritmo TF-IDF original. Se basa esencialmente en métodos estadísticos probabilísticos para aprender el significado de la distribución de categorías. Entonces, se propone el uso de la frecuencia de correlación (RF) en lugar de IDF para mejorar el reconocimiento de la clasificación de texto, es un nuevo algoritmo de ponderación de características TF-IDF, que tiene en cuenta otros factores que afectan las características. Es un nuevo método de ponderación de características, IF-IDF, en el que se agrega una nueva ponderación para representar la diferencia entre clases según el TF-IDF original.

De acuerdo con los autores Trstenjak, Bruno, Sasa y Dzenana (2014), KNN (K-Nearest Neighbors) es un algoritmo muy popular para la categorización de textos. Esta investigación presenta la posibilidad de utilizar KNN con el método de TF-IDF y un *framework* para la clasificación de textos. El *framework* está diseñado para permitir la clasificación y medición de la similitud de documentos Con base en la muestra de texto requerida. Consiste en varios módulos que llevan a los usuarios a través del proceso de clasificación. La implementación del *framework* se realiza en un entorno de desarrollo orientado a objetos con el lenguaje de programación C#.

Según los autores Doria, Montejano e Inchaurredo (2014), en este trabajo se presentó información técnica sobre repositorios abiertos (RA) temáticos y las características que estos deben reunir para garantizar la gestión de la información. Este conocimiento se transforma en el desarrollo de un modelo de un repositorio abierto. El modelo se estructura en cinco puntos de referencia: finalidad, funciones, servicios, políticas y procesos. Para llevar a cabo la construcción del repositorio que propone el modelo, se combina la metodología para el análisis, diseño y desarrollo de servicios de información digital con la minería de texto para definir los procesos de gestión documental, y la cual se dividió en fases: fase exploratoria, fase descriptiva y fase proyectiva.

Para las autoras Pérez Abelleira y Cardoso (2011), la clasificación de documentos de texto es una aplicación de la minería de textos que pretende extraer información de texto no estructurado asignando al documento una o más categorías, etiquetas o clases, basadas en el contenido. Este trabajo se centra en desarrollar técnicas para la categorización automática de documentos según su contenido, aplicando aprendizaje automático de la minería de texto. Una de las técnicas desarrolladas es un buscador semántico que aproveche el resultado de algoritmos de aprendizaje para la clasificación de documentos haciendo uso de metadatos obtenidos automáticamente por los anotadores y categorías asignadas por modelos aprendidos. Para poner en marcha su prototipo se utilizó la API Semantic Search, que ofrece una interfaz básica para el buscador semántico.

De acuerdo con el autor Yolis, Britos y Perichisky (2008), la categorización automática de documentos ha estado recibiendo creciente atención debido al aumento en la cantidad de información disponible en forma virtual y a la necesidad cada vez mayor de encontrar la información buscada en un tiempo mínimo. Si bien existen numerosos algoritmos para categorizar documentos, todos ellos evalúan un subconjunto pequeño del espacio de posibles soluciones. Aquí se presenta un algoritmo genético adaptado al problema de categorización de documentos. El algoritmo propuesto introduce cinco nuevos operadores, diseñados específicamente para la resolución del problema de categorización. Los resultados obtenidos demuestran que el algoritmo genético logra explorar el espacio de búsqueda más amplia y eficientemente que los algoritmos previos tomados como referencia. Este algoritmo presenta un nuevo operador de cruce y cuatro nuevos operadores de mutación diseñados específicamente para el problema de la categorización automática de documentos.

Esta propuesta consiste en un sistema de repositorio digital de tesis con categorización automática (sistema RT) sobre plataforma web donde se puedan almacenar investigaciones de tesis y categorizarlas automáticamente. Así también, incluir imágenes, vídeos e incluso la participación de otros usuarios. Además, las consultas

podrán seguir trabajando de manera habitual incluso con la ausencia de conexión a Internet.

Material y métodos

El desarrollo de un *software* incluye las etapas genéricas de análisis, diseño, implementación y prueba; estas etapas correctamente aplicadas garantizan un *software* desarrollado de acuerdo con los requerimientos (Pressman, 2010). A continuación se describe cada una de las etapas mencionadas.

a) Análisis del sistema

La tarea del análisis es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del *software*, y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo (Pressman, 2010), retomando el problema principal de preservar las tesis clasificadas automáticamente y hacerlas más accesibles desde un repositorio digital. Por tanto, es necesario un sistema web con acceso público y privado para publicación de las investigaciones. El sistema web es un repositorio enfocado a tesis elaboradas por estudiantes.

b) Diseño del sistema

El diseño es la primera de las tres actividades técnicas que implica un proceso de ingeniería de *software*; estas etapas son diseño, codificación y pruebas. Generalmente la fase de diseño produce un diseño de datos y un diseño arquitectónico (Pressman, 2010). A continuación se explica la arquitectura que se diseñó para la aplicación y el diagrama entidad-relación para el modelado de los datos.

b.1. Arquitectura

La arquitectura del *software* es el diseño de más alto nivel de la estructura de un sistema. Consiste en un conjunto de abstracciones que forman el “marco” del *software*. La arquitectura se diseña en la fase posterior a la de requisitos, la llamada fase de diseño (Pressman, 2010).

La arquitectura basada en capas es un estilo de programación donde el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades; el rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada (Pressman, 2010).

El estilo de arquitectura basado en capas se identifica por las siguientes características:

- Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.
- Las capas de una aplicación pueden residir en la misma máquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).
- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.

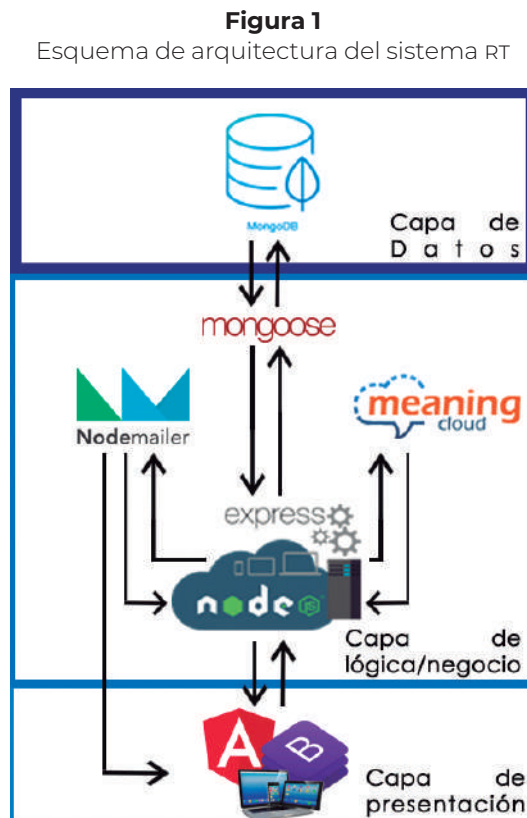
Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella. A cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

En este proyecto se implementa la arquitectura Modelo Vista Controlador, mejor conocida como “MVC”, la cual es seccionada en tres capas con el fin de dividir el sistema de la siguiente forma:

1. Capa de persistencia (datos). Esta capa se encarga de guardar todos los datos. Es donde se gestiona todo lo relativo a base de datos.

2. Capa de negocio (lógica). Esta capa se encarga de gestionar la lógica de la aplicación. Es donde se dice qué hacer con los datos.
3. Capa de presentación (usuario). En esta capa se crea la interfaz de usuario. Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

En la figura 1 se muestra el esquema de arquitectura del sistema RT.



Fuente: Elaboración propia.

Capa de datos

Esta capa está implementada con MongoDB, es importante mencionar que es una base de datos catalogada como no SQL, es orientada a documentos u objetos, esto quiere decir que en lugar de guardar datos en registros, los datos serán guardados en documentos, estos documentos son almacenados en BSON, que es una representación binaria de JSON, además de que este tipo de BD no son precisamente estructuradas, por consecuencia, son mucho más escalables (MongoDB, 2020)

Una de las diferencias más importantes con respecto a las bases de datos relacionales es que no es necesario seguir un esquema. Los documentos de una misma colección —concepto similar a una tabla de una base de datos relacional— pueden tener esquemas diferentes, lo que las hace mejores a las SQL porque son más flexibles y utilizan menos recursos.

Capa lógica

Esta capa se divide en cuatro subcapas principales:

- La primera subcapa ejecuta todas las operaciones y la comunicación con las otras dos subcapas, en esta se utiliza NodeJS (Nodejs, 2020) en conjunto con Express (Expressjs, 2020) y es denominada Servidor.
- La segunda capa es Mongoose, y es un marco de JavaScript que se usa comúnmente en NodeJS, esto ayuda a predefinir los objetos con un esquema que se asigna a un documento de MongoDB.
- La tercera subcapa se ejecuta solo cuando el servidor (la primera subcapa) le “llama”, esta subcapa se encarga de categorizar el texto o documento enviado por el servidor y responderle para que este pueda tomar decisiones con la respuesta recibida. En esta subcapa está asignada la API Meaning Cloud, que procesará los datos para la categorización de los documentos. La API Meaning Cloud categoriza los textos automáticamente en una

clasificación jerárquica o taxonomía, puede clasificar cualquier tipo de texto desde páginas web a contenido de redes sociales, de cualquier longitud y en varios idiomas (Meaning Cloud, 2020).

- La cuarta y última subcapa se encarga del envío de correos a la capa de presentación, se hace uso de un servicio externo llamado NodeMailer que permite realizar la acción de mandar un correo y llevar el control de los errores; esta subcapa es accionada solo si el servidor la llama y es capaz de comunicarle al servidor su respuesta tanto como a los diferentes clientes de la capa de presentación.

Capa de presentación

En esta capa se crea la interfaz del usuario. Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

Para lograr una mayor escalabilidad se utiliza el *framework* Angular (Angular, 2020), el cual permite desarrollar una aplicación por el lado de *front-end* más dinámica; además de Angular, se utiliza el *framework* Bootstrap (Bootstrap, 2020), el cual facilita la maquetación de sitios web, además de ser compatible con procesadores.

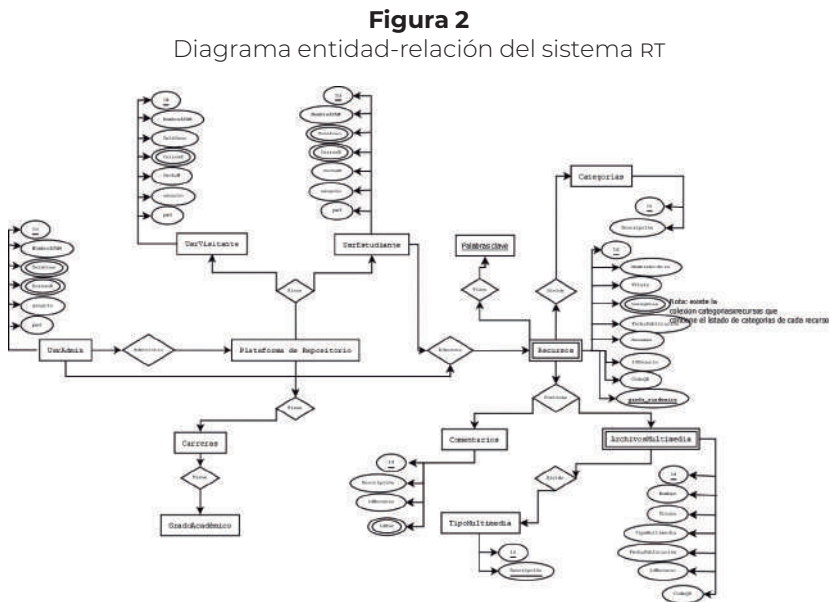
En la capa de presentación se realizan las siguientes actividades: a) inicio de sesión, b) control de usuarios (administrador, estudiante, visitante), c) control de recursos (documentos, fotos, vídeos), d) control de comentarios, e) control del foro, f) categorización de recursos.

Las tecnologías que se usan en las diferentes capas en conjunto forman la tecnología conocida como Stack MEAN (Mongo, Express, Angular, Node.JS), reconocida en su mayoría por ser un conjunto de componentes de código abierto que proporcionan un marco de extremo a extremo para crear aplicaciones web dinámicas; desde la parte superior (código que se ejecuta en el navegador) hasta la parte inferior (base de datos).

b.2. Diagrama entidad-relación

Un diagrama entidad-relación es un diagrama de flujo que ilustra cómo las “entidades” (personas, objetos, etc.) se relacionan entre sí dentro de un sistema. Usualmente se utilizan para diseñar o filtrar bases de datos y fueron creados para facilitar el diseño de arquitecturas haciendo uso de distintos símbolos, como son: rectángulos, flechas, óvalos, líneas de conexión, entre otras cosas (Lucid, 2020).

En la figura 2 se muestra el diagrama entidad-relación que se diseñó para el posterior desarrollo del sistema RT.



Fuente: Elaboración propia.

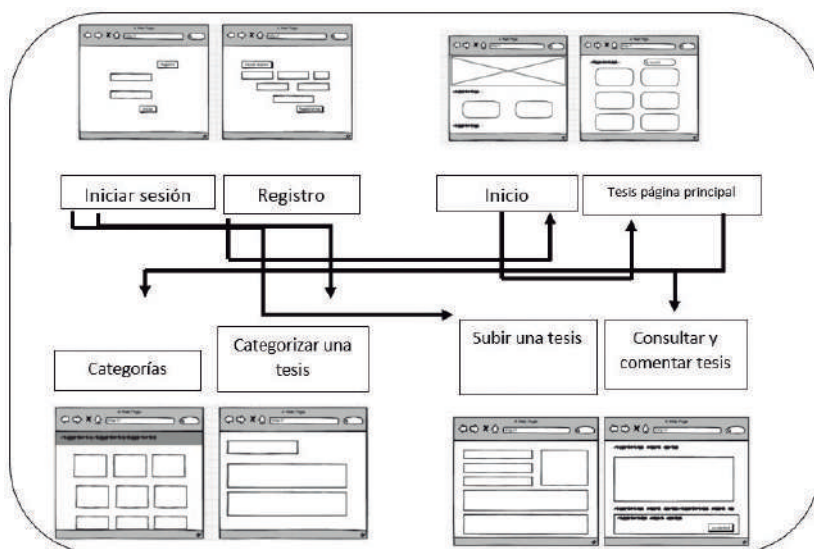
Cada uno de los rectángulos representan las tablas en la base de datos (UsrVisitante, UsrEstudiante, UsrAdmin, Categorías, etc.), los óvalos representan los atributos/datos que necesitan las entidades y que el sistema solicitará al momento de hacer algún registro o consulta.

Por último, los rombos representan las conexiones o relaciones que existen entre una entidad y otra.

a) Implementación (interfaces)

Como se muestra en la figura 3 se pueden identificar las distintas interfaces con las que cuenta el sistema RT y cómo es que se relacionan entre sí.

Figura 3
Interfaces del sistema RT



Fuente: Elaboración propia.

Para la implementación de la aplicación web se emplean diversas tecnologías para la categorización automática de las tesis de los estudiantes y participantes en general, que decidan subir sus documentos al repositorio. Las tecnologías que se utilizaron con la ayuda del Modelo Vista Controlador (MVC) se dividen en capas de la siguiente manera:

Modelo: Utilizamos MongoDB, ya que es una base de datos no relacional y es dinámica a los datos que se van a almacenar. Guarda los datos en formato BSON, lo que quiere decir que almacena los datos en formato JSON binario, ayudando a economizar espacio y recurso, además de poder realizar las consultas de manera más rápida. Al ser una base de datos totalmente en lenguaje JavaScript, ayudó con la utilización de un solo lenguaje, para todas las capas.

Controlador: El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU / Linux, etc.), Microsoft Windows, Macintosh y otros. El sistema de repositorio de tesis con categorización automática se instaló e implementó en el servidor que está alojado en el sistema operativo CentOS de la UAS. El servidor se ejecutó desde node.js en el mismo sistema operativo y la aplicación se ejecutó en un servidor independiente de node.js, la aplicación es capaz de ejecutarse en el 90 % de los navegadores web más utilizados.

Vista: Bootstrap es una biblioteca multiplataforma o un conjunto de herramientas de código abierto para diseñar sitios web y aplicaciones. Contiene plantillas de diseño con tipografía, formularios, botones, tablas, menús de navegación y otros elementos de diseño basados en HTML y CSS, así como extensiones adicionales de JavaScript. Además de proporcionar dinamismo al sitio, se utiliza jQuery, es una biblioteca JavaScript multiplataforma que simplifica la forma de interactuar con documentos HTML, manipula el árbol DOM, maneja eventos, desarrolla animaciones y agrega interacción con la tecnología AJAX a los sitios web.

El sistema de repositorio digital de tesis con categorización automática (sistema RT) es una plataforma en donde cualquier persona con interés en conocimiento puede subir o consultar trabajos académicos de una persona que ya haya subido su trabajo con anterioridad al sistema. La figura 4 presenta la vista principal en donde se podrán realizar búsquedas.

Desde esta misma página web también se puede ir a la vista principal, solicitar ayuda para informarse sobre lo que se puede hacer en

el repositorio, o bien se puede iniciar sesión para ingresar al repositorio o registrarse para ingresar al sistema.

El sistema cuenta con distintos tipos de usuarios adaptándose a las necesidades de cada uno y asegurando los datos y documentos de los usuarios; los usuarios que existen son cuatro: un visitante, un usuario registrado, un estudiante y un administrador, en donde cada uno tiene distintas prioridades.

Figura 4

Vista principal del sistema RT para realizar búsquedas



Fuente: Elaboración propia.

El visitante puede consultar las tesis que están disponibles y descargarlas. Pueden comentar las tesis que están disponibles en la plataforma, para ello deberán registrarse como visitantes y se les habilitará la opción de suscripción al autor del documento. El estudiante puede consultar las tesis que están disponibles y descargarlas. Pero, al ser estudiante de la escuela, cuentan con un número único, con él serán dados de alta por el administrador para que posteriormente pue-

dan subir una tesis. En cuanto a las tesis que subirán los estudiantes, estos tienen la posibilidad de subirlas con archivos multimedia como videos, fotos/imágenes, y pueden añadir palabras clave dentro de sus tesis para complementar al categorizador automático y hacer una asignación más acertada del documento.

El sistema con un estudiante en la sesión le mostrará una vista distinta a la de un visitante, este subirá su tesis llenando el formulario y el sistema de manera automática identificará términos con pesos positivos y negativos que haya asignado el estudiante, relevantes e irrelevantes dentro del texto, y así logrará categorizar de manera efectiva. Además, cuenta con un campo de entrenamiento para que el estudiante pueda colocar un fragmento del documento en donde el categorizador identificará de manera más amplia los términos asignados por el estudiante anteriormente dentro de dicho fragmento.

El administrador, además de poder realizar todas las acciones anteriores, puede registrar y eliminar un estudiante, consultar todos los estudiantes, realizar control de facultades/instituciones, dar de baja y modificar sus datos, agregar tesis directamente, consultar las tesis activas, y ver las tesis que están sin aprobar, lo que indica que aún no están disponibles en la plataforma; tendrá un control de las solicitudes de tesis que los estudiantes quieran subir, para evitar que se suban documentos “basura”, así podrá ser más confiable la documentación que esté activa en la plataforma, y las tesis que aún no son categorizadas, que estas últimas tampoco están disponibles y son cargadas en esta sección al momento en el que se sube una nueva tesis y no existe una conexión de Internet o es inestable, no presentará ningún problema el sistema, pero almacenará esas tesis hasta que sean categorizadas, lo que quiere decir que hasta que exista una conexión a Internet. Además de agregar categorías nuevas para la categorización automática y consultar las que ya están activas, podrá dar de baja una tesis que esté activa y eliminar los comentarios que estén en las tesis activas que fueron comentadas de una manera inadecuada.

Tendrá el control de las tesis que se publican y se muestran en la página principal de la plataforma, señalando cuáles son las más

vistas, las categorías a mostrar, las tesis más recientes y los datos de contacto, en donde él puede modificar los datos.

a) Pruebas (evaluación)

Después de realizar análisis cuantitativos y revisar datos del comportamiento de los usuarios al usar el repositorio de tesis universitarias, se decidió usar el System Usability Scale (SUS). El sistema de escala de usabilidad es una herramienta metodológica que se usa para medir la usabilidad de un objeto, dispositivo o aplicación. La escala en sí consiste en diez preguntas, cada una de las cuales puede ser puntuada de 1 a 5, donde 1 significa total desacuerdo y 5 significa total acuerdo (Brooke, 1996).

Aunque esta escala es extraordinariamente simple de usar, diferentes pruebas y tests han demostrado que los resultados obtenidos a partir de ella suelen ser muy confiables y acertados, razón por la cual es uno de los métodos de medición de usabilidad más utilizados en experiencia de usuario (Brooke, 1996).

Técnica de muestreo

Se utilizó la técnica de muestreo no probabilística, en donde los sujetos generalmente son seleccionados en función de su accesibilidad o criterio personal e intencional del investigador. Se decidió aplicar la técnica de muestreo no probabilística debido a su accesibilidad y fácil uso, además de la libertad que brinda para seleccionar problemas específicos.

Participantes y muestra

De acuerdo con Nielsen (2000), se pueden realizar pruebas de usabilidad con tan solo cinco personas, esto ha sido ampliamente aceptado debido a que con esta cantidad de usuarios es suficiente para revelar hasta en un 85 % los problemas de usabilidad. En este contexto,

se seleccionaron cinco usuarios de manera aleatoria. Los usuarios seleccionados fueron docentes de la Facultad de Ingeniería Mochis (FIM) de la Universidad Autónoma de Sinaloa (UAS) enroladas en el posgrado de la UAS, con la experiencia suficiente para brindar opiniones positivas en el repositorio de tesis. En la tabla 1 se presentan los detalles y perfiles de los participantes.

Tabla 1

Detalle perfil de los participantes

Género	Edad	Ocupación	Tipo de usuario	Uso de Internet	Dispositivos
Mujer	30 aprox.	Doctora en Ciencias	Investigador/ Visitante	Activo	Computadora
Hombre	30 aprox.	Doctor en Ciencias	Investigador/ Visitante	Activo	Computadora
Hombre	50 aprox.	Doctor en Ciencias	Investigador/ Visitante	Activo	Computadora
Hombre	50 aprox.	Doctor en Tecnología	Investigador/ Visitante	Activo	Computadora
Hombre	40 aprox.	Doctor	Administrador	Activo	Computadora

Fuente: Elaboración propia.

Las métricas que se utilizaron para analizar el comportamiento de los participantes con el uso del sistema RT fueron:

1. Sistema de Escala de Usabilidad (SUS)
2. Tiempo de la prueba
3. Éxito de realización de la prueba
4. Calificación de nivel de dificultad

Metodología

El procedimiento para evaluar el sistema RT fue el siguiente:

- Primero, se proporcionó el enlace de acceso al sitio RT a los usuarios participantes: visitante y administrador. Segundo, se le proporcionó la clave de acceso a uno de los participantes como administrador. Tercero, se aplicó una prueba con cuatro tareas

y escenarios que fueron leídas y asignadas a cada usuario del sistema:

1. Primera tarea, el usuario administrador dio de alta a cada visitante para que pudieran acceder como tal, dentro del repositorio de tesis. El resto de los usuarios se registraron como visitantes.
2. Segunda tarea. El administrador dio de alta una categoría para poder asignar una tesis a dicha categoría.
3. Tercera tarea. Los usuarios visitantes dieron de alta una nueva tesis desde su perfil con fotos y vídeos.
4. Cuarta tarea. Se registraron los usuarios como visitantes y les llegó un mensaje al correo electrónico que dieron de alta durante su registro, se suscribieron a un estudiante que subió una tesis y dicho estudiante subió una nueva tesis; posteriormente, les llegó un nuevo correo donde se notifica al visitante que el estudiante al que están suscritos subió una nueva tesis.

Cuarto, por cada tarea y escenario se tomó el tiempo de la prueba, se comprobó el éxito de realización de la prueba y desde la percepción del usuario se calificó el nivel de dificultad. Quinto, se aplicó la encuesta de usabilidad SUS. Finalmente, se abrió una sesión de preguntas y sugerencias por los usuarios participantes para la mejora del sistema RT.

La sesión duró cerca de 1 hora con 20 minutos, desde que se inició con el acceso al sistema RT hasta la sesión de preguntas y sugerencias.

Resultados

En la tabla 2 se pueden observar los resultados de las encuestas realizadas. Cada renglón en la tabla representa una encuesta contestada por el visitante, en este caso cinco encuestas contestadas. Las colum-

nas con la etiqueta P y un número asociado hacen referencia a las diez preguntas de la encuesta de usabilidad SUS.

Tabla 2
Datos de encuesta de usabilidad sus

# Visitantes	Preguntas de la encuesta sus									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Usuario 1	5	2	4	3	4	1	5	1	5	1
Usuario 2	5	1	4	1	4	1	4	1	4	1
Usuario 3	5	1	5	2	5	1	5	1	5	1
Usuario 4	5	2	4	1	4	1	4	1	5	1
Usuario 5	5	2	4	3	4	1	5	1	5	1

Fuente: Elaboración propia.

Como se puede apreciar en la tabla 2, en cada pregunta las opciones de respuesta son: valor de (1) que representa a “totalmente en desacuerdo” a valor de (5) que representa a “totalmente de acuerdo”. Luego, estos datos se procesan, por lo que la puntuación de cada pregunta oscilará entre 0 y 4. Para ello se aplica la siguiente regla:

- Para las preguntas impares la puntuación es la posición de escala menos 1.
- Para las preguntas pares la puntuación es 5 menos la posición de la escala.

Finalmente, estos puntajes deben sumarse por encuesta-participante y luego multiplicarlos por 2.5 para obtener el valor general de SUS. Como resultado, las puntuaciones deben alcanzar un rango de 0 a 100 puntos SUS, los cuales deben normalizarse para obtener una interpretación final del resultado (Brooke *et al.*, 1996). Los datos procesados en esta investigación se pueden apreciar en la tabla 3.

Tabla 3

Datos de encuesta procesados

# Visitantes	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Puntuación sus
Usuario 1	4	3	3	2	3	4	4	4	4	4	87.5
Usuario 2	4	4	3	4	3	4	3	4	3	4	90
Usuario 3	4	4	4	3	4	4	4	4	4	4	97.5
Usuario 4	4	3	3	4	3	4	3	4	4	4	90
Usuario 5	4	2	4	4	4	0	4	4	4	4	85
Puntuación promedio = 90 puntos sus											

Fuente: Elaboración propia.

En la tabla 3 se puede apreciar que la puntuación promedio es de 90. Para normalizar la puntuación se pueden considerar los criterios de la tabla 4 que plantea el sitio web MeasuringU (empresa de investigación que se centra en el análisis estadístico del comportamiento humano) (Sauro, 2018).

Tabla 4

Grados, adjetivos y aceptabilidad asociadas con puntajes brutos del sus

Puntuación sus	Rango porcentual	Adjetivos	Aceptabilidad
84.1-100	96-100	Mejor imaginable	Aceptable
80.8-84.0	90-95	Excelente	Aceptable
78.9-80.7	85-89		Aceptable
77.2-78.8	80-84		Aceptable
74.1-77.1	70-79		Aceptable
72.6-74.0	65-69		Aceptable
71.1-72.5	60-64	Bueno	Aceptable
65.0-71.0	41-59		Marginal
62.7-64.9	35-40		Marginal
51.7-62.6	15-34	OK	Marginal
25.1-51.6	2-14	Pobre	No aceptable
0-25	0-1.9	Peor imaginable	No aceptable

Fuente: Adaptado del sitio web *MeasuringU* (Sauro, 2018).

De acuerdo con la información de la tabla 4, las puntuaciones SUS en el rango de 84.1-100 caen en el rango porcentual de usabilidad de 96 %-100 % con el adjetivo para describir la experiencia de usuario como la “Mejor imaginable” y “Aceptable”.

Por otra parte, la realización de las tareas fue favorable de acuerdo al promedio SUS alcanzado, la usabilidad del sistema RT es bastante buena. Para resumir el resultado de la ejecución de las tareas se puede decir que los resultados por cada tarea son:

Tarea uno, se lograron registrar todos los tipos de usuario correctamente sin problema. Tarea dos, se logró registrar la categoría con sus palabras clave, sin embargo, dicha categoría presentó un error al querer utilizarse por cuestiones de red. Tarea tres, se logró dar de alta a una nueva tesis con archivos multimedia (fotos y vídeo) correctamente. Tarea cuatro, llegaron los correos a los usuarios al momento de registrarse. También, llegaron los correos al momento de que un estudiante (al que están suscritos) subiera una nueva tesis con éxito.

En todas las tareas:

1. Los tiempos para realizar las tareas eran cortos considerablemente.
2. Los usuarios participantes eran inexpertos en el sistema, y aun así lograron comprender el manejo y funcionamiento de la plataforma.
3. Es una plataforma intuitiva para los participantes de edades variadas y no hubo problemas con su flujo de funcionamiento habitual.
4. La calificación de nivel por dificultad fue de 90/100 según los resultados de la escala de usabilidad SUS.

En la tabla 5 se describe el porcentaje de éxito que asignó cada uno de los participantes a las tareas durante la prueba que realizaron al sistema RT.

Tabla 5
Puntaje de éxito por participantes

Participante	Tarea 1	Tarea 2	Tarea 3	Tarea 4
Participante 1	100 %	0	100 %	100 %
Participante 2	100 %	100 %	100 %	100 %
Participante 3	100 %	100 %	100 %	100 %
Participante 4	100 %	100 %	100 %	100 %
Participante 5	100 %	100 %	100 %	100 %
Tasa media de éxito por Tarea	100 %	80 %	100 %	100 %

Fuente: Elaboración propia.

En la tabla 6 que se presenta a continuación se describen los errores que surgieron durante la prueba que realizaron los participantes al sistema RT; cabe mencionar que los errores que se presentaron ya fueron resueltos.

Tabla 6
Errores registrados por tarea

Tarea	Descripción	Número de veces
Tarea 1	Sin errores	n/a
Tarea 2	Falló al utilizar categoría registrada	1
Tarea 3	Sin errores	n/a
Tarea 4	Sin errores	n/a

Fuente: Elaboración propia.

En la tabla 7 se presentan las recomendaciones de los participantes para la complementación de la plataforma que se desarrolló.

Tabla 7
Observaciones por participantes con orden prioritario

Observaciones	Interpretación	Prioridad (1 a 5)	Recomendación
Que los visitantes también puedan subir una tesis.	Todos los usuarios podrán registrarse y solo el administrador les otorgará el permiso de poder subir una tesis.	5	El administrador podrá cambiar el estatus de un visitante al de un estudiante.
¿Qué pasa si el administrador comete un error al subir una tesis?	Que el administrador pueda eliminar y editar cualquier dato de una tesis.	5	El administrador podrá editar, eliminar y recategorizar una tesis en caso de ser necesario.
Dividir las tesis por nivel de estudio y licenciatura.	Que la información que describe la tesis se adecue a los estándares que existen actualmente en los repositorios.	5	Basar la consulta de tesis en otros repositorios ya existentes.
Tener una pantalla donde se muestre información general de la tesis al usuario visitante antes de entrar a ver el documento PDF.	Que se puedan visualizar de manera más clara y concisa los datos generales de la tesis sin necesidad de abrir el documento.	4	Que el usuario pueda ver solo los datos de interés sin abrir el documento.
Que una vista de todas las tesis pueda ser visualizada en forma de lista.	Que el usuario pueda tener la opción de poder visualizar las tesis de distintas maneras.	3	Que se pueda modificar la vista de tesis para facilitar consulta.

Fuente: Elaboración propia.

Discusión y conclusiones

Discusión

La métrica que se utilizó para analizar el comportamiento de los usuarios fue la escala de usabilidad SUS, debido a que es extraordinariamente simple de usar y diferentes pruebas han demostrado que los resultados obtenidos a partir de ella suelen ser muy confiables y acertados.

El tiempo que duró la realización de la prueba fue de 1 hora 20 minutos, la tarea número 1 duró un aproximado de 15 minutos, la tarea número 2 duró 15 minutos, la tarea número 3 duró un tiempo aproximado de 20 minutos y la tarea número 4 duró 15 minutos aproximadamente. La escala de usabilidad SUS que consiste en una encuesta de diez preguntas se contestó al final de la sesión con una duración de aproximadamente 15 minutos, el resto del tiempo se ocupó en sugerencias para mejoras del sistema RT.

Cada tarea fue completada exitosamente, y los participantes se mostraron satisfechos con la funcionalidad del repositorio. La calificación que la métrica de usabilidad SUS alcanzó fue de 90 puntos, muy por arriba del promedio general aceptable (70 puntos SUS); estos resultados indican que el sistema RT utilizando adjetivos es el “Mejor imaginable”.

Los resultados obtenidos en la ejecución de las tareas demuestran que el sistema es capaz de ejecutarlas sin problema. No obstante, surgió un fallo al categorizar con una nueva categoría (tarea número 2) debido a que no se contaba con las suficientes palabras clave y no se tenía texto de entrenamiento. Posterior a las tareas se añadió un texto de entrenamiento y este problema se resolvió.

Los resultados de la prueba de la escala de usabilidad SUS muestran que el sistema es usable, sin embargo, puede mejorar. Las observaciones que los participantes realizaron brindaron un gran apoyo sobre cómo mejorar el sistema RT en la presentación de las tesis, y los criterios de búsqueda.

Conclusiones

El proyecto que se desarrolló fue una experiencia satisfactoria y de mucho conocimiento para el área de ingeniería de *software*. Se alcanzó el objetivo principal de desarrollar un sistema de repositorio digital de tesis con categorización automática, denominado sistema RT, para hacer frente al problema de preservar las tesis digitales de licenciatura y posgrado de una institución educativa y con ello hacer accesibles los resultados de investigación.

Además, es importante decir que el sistema RT fue desarrollado con base en las etapas genéricas de análisis, diseño, implementación y prueba del *software* planteadas por Pressman (2010). Por otra parte, se utilizaron tecnologías nuevas y de alta mantenibilidad, denominadas Stack MEAN (Mongo, Express, Angular, Node.JS), y la API Meaning Cloud. Además, el uso de estas tecnologías no tuvo ningún costo.

Los resultados de las pruebas de la escala de usabilidad SUS fueron las esperadas, debido a que presentó un puntaje de 90/100 puntos SUS. Los usuarios participantes que probaron el sistema indicaron que no era tan complicada su utilización y estuvieron conformes con la categorización automática. Esta fue una de las características que más llamó la atención ya que, como se mencionó en secciones anteriores, no se dispone con repositorios de tesis con esta característica.

Con el desarrollo de este repositorio se puede aumentar el número de estudiantes interesados en la investigación, ya que podrán tener acceso a más conocimiento (documentos de tesis) de otros usuarios y se motivarán a realizar sus propios proyectos de investigación que podrán compartir.

Con una visión a corto plazo, se espera que se le dé continuidad a esta investigación por estudiantes de la comunidad universitaria. Las áreas de oportunidad son las siguientes: a) incluir la lectura de los metadatos de los archivos multimedia para incluirlos en la clasificación; b) diseñar y desarrollar un clasificador propio, para no depender de la API Meaning Cloud.

Referencias

- Angular (2020), “Angular”, *Angular*, 1 de agosto. Disponible en: <https://angular.io/>
- AngularFramework (2020), “One Framework Mobile & Desktop”, *One framework mobile & desktop*, 1 de agosto. Disponible en: <https://angular.io/>
- Balakrishna, C. (2012), “Enabling Technologies for Smart City Services and Applications”, *Sixth International Conference on Next Generation Mobile Applications, Services and Technologies*, pp. 223-226, Cardiff.
- Bayardo, R. (1998), “Efficiently Mining Long Patterns from Databases”, *ACM-SIGMOD Int’l Conf. on Management of Data*, pp. 85-93. Consultado: septiembre 18, 2019.
- Bootstrap (2020), “Build Fast, Responsive Sites with Bootstrap”, *Build Fast, Responsive Sites with Bootstrap*, 1 de agosto. Disponible en: <https://getbootstrap.com/>
- Brooke, J. (1996), “sUS-A Quick and Dirty Usability Scale”, *Usability Evaluation in Industry*, 189(194), pp. 4-7.
- Chichernea, V. (2014), “The Use of Decision Support Systems (DSS) in Smart City Planning and Management”, *Romanian Economic Business Review, Romanian-American University*, pp. 238-251.
- Desouza, K. & A. Bhagwatwar (2012), “Citizen Apps to Solve Complex Urban Problems”, *Journal of Urban Technology*.
- Doria, M. V.; Montejano, G. A. y C. I. Inchaurredo (2014), “Modelo de repositorio abierto en Ingeniería de Software para el ámbito universitario”, *XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO) - I Simposio Argentino de Tecnología y Sociedad (STS)*, pp. 125-139, Argentina, Buenos Aires.
- Escobar Acevedo, A. (2009), *Métodos de refinamiento de la clasificación translingüe de documentos (Tesis de maestría)*, septiembre, Puebla, Tonantzintla, Instituto Nacional de Astrofísica, Óptica y Electrónica INAOE.
- ExpressFramework (2020), “Express infraestructura web rápida, minimalista y flexible para Node.js”, *Express infraestructura web rápida, minimalista y flexible para Node.js*, 1 de agosto. Disponible en: <https://expressjs.com/es/>
- Expressjs (2020), “Express”, *Express*, 1 de agosto. Disponible en: <https://expressjs.com/es/>

- Fan, H. y Y. Qin (2018), *Research on Text Classification Based on Improved TF-IDF Algorithm*, pp. 501-506, Atlantis Press, DOI:<https://doi.org/10.2991/ncce-18.2018.79>
- Fowler, M. y K. Scott (1999), *UML gota a gota*, Alhambra Mexicana.
- GmbH, d. (2020), “DSpace”, *DSpace*. Disponible en: <https://www.dspace.com/en/pub/home.cfm>. Consultado: enero 6, 2020.
- Hernández Sampieri, R.; Fernández Collado, C. y M. Baptista Lucio (2014), *Metodología de la Investigación*, 6ª ed., McGraw-Hill.
- IonicFramework (2020), “Ionic Community”, *Ionic Community*, 1 de agosto. Disponible en: <https://ionicframework.com/community>
- Jisc (2020), “Jisc”. Disponible en: http://http://v2.sherpa.ac.uk/view/repository_by_country/Mexico.html. Consultado: febrero 4, 2020.
- Kirabowski, J. & M. Corbett (1993), “SUMI: The Software Usability Measurement Inventory”, *British Journal of Educational Trchnology*, 24(3), pp. 210-212.
- Lemos, A. y N. Veras (2018), “Cidadão Sensor e Cidade Inteligente: Análise dos Aplicativos Móveis da Bahia”, *Revista Famecos*.
- Lucid, S. (2020), “Lucidchart”. Disponible en: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>. Consultado: febrero 12, 2020.
- Maciá Pérez, F. (2017), *Smart University - Hacia una universidad más abierta*, Alfaomega, Marcombo.
- Meaning Cloud (2020), “Meaning Cloud”, Meaning Cloud, 1 de agosto. Disponible en: <https://www.meaningcloud.com/es>
- Medina González, A. (2017), “Implementación de un repositorio digital para el entorno local de la Facultad de Comunicación de la Universidad de la Habana”, 13(2), pp. 202-214.
- MongoDB (2020), “MongoDB”, *MongoDB*, 1 de agosto. Disponible en: <https://www.mongodb.com/es>
- N. Ocholla, D. (2011), “An Overview of Issues, Challenges and Opportunities of Scholarly Publishing in Information Studies in Africa”, *African Journal of Library, Archives & Information Science*, 21(1), pp. 1-16.
- Nodejs (2020), “Node.js”, *Node.js*, 1 de agosto. Disponible en: <https://nodejs.org/es/>
- (2020), “Node.js® es un entorno de ejecución para JavaScript”, *Node.js® es un entorno de ejecución para JavaScript*, 1 de agosto. Disponible en: <https://nodejs.org/es/>

- Parra-Valeancia, J.; Guerrero, C. y D. Rico (2017), "IoT: Una aproximación desde ciudad inteligente a universidad inteligente", *Revista Ingenio UFPSO*, 13, pp. 9-15.
- Pérez Abelleira, M. A. y A. C. Cardoso (2011), "Categorización automática de documentos", *Simposio Argentino de Inteligencia Artificial*, pp. 20-31, Argentina, Salta.
- Pressman, R. (2010), *Ingeniería de software. Un enfoque práctico*, 7ª ed., México D.F., Editorial McGraw-Hill.
- Sommerville, I. (2005), *Ingeniería del software*, 7ª ed., Madrid, Pearson.
- Southampton, U. o. (2020), "eprints.com". Disponible en: <https://www.eprints.org/uk/>. Consultado: marzo 20, 2020.
- Texier, J.; De Giusti, M. R.; Oviedo, N. F.; Villarreal, G. L. y A. J. Lira (2002), "El uso de repositorios y su importancia para la educación en Ingeniería", *World Engineering Education Forum (WEEF)*, Argentina, Buenos Aires.
- Trstenjak; Bruno, M.; Sasa, D. y Dzenana (2014), "KNN with TF-IDF Based Framework for Text Categorization", *24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013*, pp. 1356-1364, Eslovenia, Elsevier, DOI: 10.1016/j.proeng.2014.03.129
- Yolis, E.; Britos, P. y Perichisky (2008), "Algoritmos genéticos aplicados a la categorización automática de documentos", *Revista Eletrônica de Sistemas de Informação*, 2(2), pp. 1-15, DOI: 10.21529/RESI.2003.0202001

Tecnologías emergentes.
Un compendio de desarrollos tecnológicos
se terminó de editar en diciembre de 2020
en los talleres de Ediciones de la Noche
Madero #687, Zona Centro
44100, Guadalajara, Jalisco, México.

El tiraje fue de 1 ejemplar.

Hoy en día las tecnologías emergentes han abierto un sinnúmero de posibilidades para ser implementadas en diferentes ámbitos de la vida. Desde esta perspectiva, las tecnologías permiten ser un medio para ayudar en el desarrollo de innovaciones que permiten fortalecer procesos y toma de decisiones en las organizaciones.

En la presente obra, el lector podrá encontrar desarrollos tecnológicos en el ámbito educativo y en el ámbito de las organizaciones. Así mismo, las nuevas tecnologías permite que muchas de estas organizaciones incrementen sus proyectos con la implementación de las nuevas tecnologías para adaptarlas a sus procesos con el propósito de innovar o mejorar sus productos o servicios.

