



Revista de Investigación en Tecnologías de la Información
ISSN: 2387-0893
revista.riti@gmail.com
Universitat Politècnica de Catalunya
España

Castillo Estrada, Christian Mauricio; Cancino Villatoro, Karina;
Benavides García, Vanessa; de la Cruz Vázquez, Aron
Diseño de un Sistema web para el control de Curriculum Vitae Electrónico de
personal docente basado en una arquitectura orientada a servicios (API REST)
Revista de Investigación en Tecnologías de la Información,
vol. 10, núm. 20, 2022, Enero-Junio, pp. 28-42
Universitat Politècnica de Catalunya
España

DOI: <https://doi.org/10.36825/RITI.10.20.003>

- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org





Diseño de un Sistema web para el control de Curriculum Vitae Electrónico de personal docente basado en una arquitectura orientada a servicios (API REST)

Design of a web system for the control of the Electronic Curriculum Vitae of teaching staff based on a service-oriented architecture (REST API)

Christian Mauricio Castillo Estrada

Universidad Autónoma de Chiapas, Tapachula, Chiapas, México
cmce@unach.mx

ORCID: 0000-0002-4540-9980

Karina Cancino Villatoro

Universidad Politécnica de Tapachula, Tapachula, Chiapas, México
karina.cancino@uptapachula.edu.mx

Vanessa Benavides García

Universidad Autónoma de Chiapas, Tapachula, Chiapas, México
vanessa.benavides@unach.mx

Aron de la Cruz Vázquez

Universidad Autónoma de Chiapas, Tapachula, Chiapas, México
aron.cruz@unach.mx

doi: <https://doi.org/10.36825/RITI.10.20.003>

Recibido: Enero 14, 2022

Aceptado: Marzo 04, 2022

Resumen: El presente artículo describe la propuesta de generar un sistema de información basado en entorno web para el control de Curriculum Vitae Electrónico del personal docente de la Facultad de Negocios Campus IV de la Universidad Autónoma de Chiapas, para efectos de concentrar datos de índole laboral y de productividad académica del personal docente que imparten asignaturas en los programas educativos que oferta la institución educativa, mismos que son evaluados periódicamente por organismos acreditadores; así también, este sistema de información contempla la generación de reportes estadísticos personalizados que coadyuvan a la toma de decisiones de los directivos. Para desarrollar el sistema mencionado se utilizaron técnicas y metodologías de ingeniería web basadas en una arquitectura orientada a servicios. La solución tecnológica resultante ofrece una interfaz gráfica de usuario responsiva para adaptarse a la resolución de diversos equipos de cómputos y dispositivos móviles, para ello, se utilizó el *framework* progresivo Vue.js derivado del lenguaje JavaScript y la librería de componentes de interfaz de usuario PrimeVue. En lo relativo a la construcción del API de servicio (API REST), es decir, la programación de lado del servidor se utilizó el *framework* de código abierto Codeigniter basado en el lenguaje PHP y como gestor de base de datos MySQL.

Palabras clave: *Aplicación Web, Gestión de información, Ingeniería Web, API REST.*

Abstract: This article describes the proposal to generate an information system based on a web environment for the control of the Electronic Curriculum Vitae of the teaching staff of the Faculty of Business Campus IV of the Autonomous University of Chiapas, for the purpose of concentrating labor and productivity data academic staff of the teaching staff who teach subjects in the educational programs offered by the educational institution, which are periodically evaluated by accrediting bodies; Likewise, this information system contemplates the generation of personalized statistical reports that contribute to the decision-making of managers. To develop the aforementioned system, web engineering techniques and methodologies based on a service-oriented architecture were used. The resulting technological solution offers a responsive graphical user interface to adapt to the resolution of various computer equipment and mobile devices, for this, the progressive framework Vue.js derived from the JavaScript language and the PrimeVue user interface component library were used. Regarding the construction of the Service API (REST API), that is, the server-side programming, the Codeigniter open-source framework was used based on the PHP language and as a MySQL database manager.

Keywords: *Web Application, Information Management, Web Engineering, REST API.*

1. Introducción

El proceso de acreditación de un programa académico, tanto de técnico superior universitario, como de nivel superior, es el reconocimiento público que una organización acreditadora otorga, en el sentido de que éste cumple con determinados criterios y parámetros de calidad; en otras palabras, demuestra que el programa tiene pertinencia social, tal como lo define el organismo acreditador CACEI en su sitio web oficial [1].

La Facultad de Negocios de la UNACH, actualmente posee todos sus programas educativos acreditados por organismos acreditadores reconocidos por el Consejo para la Acreditación de la Educación Superior A.C. (COPAES) que actúa como única instancia autorizada por el Gobierno Federal a través de la Secretaría de Educación Pública (SEP), para conferir reconocimiento formal y supervisar a organizaciones cuyo fin sea acreditar programas académicos del tipo superior que se imparten en México [2].

Estos programas educativos poseen planes de mejora continua que permiten dar seguimiento a las recomendaciones emitidas durante la acreditación; para ello, es necesario integrar carpetas de evidencias documentales de manera física y electrónica que permitan solventar cada recomendación; por lo anterior se elaboró una propuesta de construir un sistema de información para concentrar la información del personal docente de forma electrónica respetando los formatos e indicadores que establecen dichos organismos acreditadores. Considerando dicha propuesta, cada docente tendría la facilidad y el acceso de capturar información relacionada con la productividad académica que genera como parte de sus funciones en los rubros de Investigación, Extensión, Vinculación y Docencia correspondiente a cierto periodo a evaluar.

La innovación tecnológica y la automatización de procesos de negocio en las organizaciones, resultan en la actualizar factores claves para elevar la competitividad y mejorar el desempeño del personal. En virtud de ello, se realizó una propuesta que tiene como objetivo desarrollar e implementar un sistema de información basado en un entorno web para el registro y control de información de productividad académica del personal docente. El aseguramiento de la calidad en las Instituciones Educativas de nivel Superior depende un trabajo colaborativo, innovador y constante que permita una mejora continua en sus diferentes procesos y áreas organizacionales; en ese sentido, la propuesta presentada resulta ser factible para coadyuvar a la consolidación de los procesos de acreditación de programas educativos.

El presente trabajo académico describe la propuesta relativa al desarrollo de un sistema de información basado en tecnologías web para el control de Curriculum Vitae Electrónico del personal docente que cumple con los formatos y lineamientos que establecen los organismos acreditadores. El documento posee una estructura que se divide de la siguiente manera: la Sección 2, expone el estado del arte; la Sección 3, detalla la metodología de desarrollo y las herramientas tecnológicas utilizadas; en lo que respecta a la Sección 4 da a conocer los resultados de las pruebas de funcionamiento efectuadas. Finalmente, la Sección 5 presenta las conclusiones y recomendaciones a futuro.

2. Estado del arte

2.1. Marco Referencial

De acuerdo con Rueda, la Conferencia Regional de la Educación Superior en América Latina y el Caribe recomendó a los gobiernos que la integran, la consolidación de sistemas nacionales de acreditación y evaluación, con la finalidad de apoyar la construcción e implementación de sistemas de autoevaluación de Instituciones de Educación Superior (IES). Así también, se les ha recomendado a las IES trabajar en la búsqueda de un continuo mejoramiento en el cumplimiento de sus misiones institucionales, fomentar la cultura de la evaluación y el compromiso con los sistemas de evaluación y acreditación [3].

Es evidente observar que uno de los grandes desafíos para las instituciones de educación superior que han decidido someter sus programas educativos a procesos de evaluación con fines de lograr una acreditación, resulta ser que la comunidad universitaria acepte el compromiso de optar una actitud positiva e impulsar una cultura de mejora continua, sin olvidar que los beneficios de dichos procesos no son, necesariamente, automáticos. Las autoridades de las IES deben estar conscientes que participar en estos procesos, implica una búsqueda de reconocimiento social y de prestigio; en este sentido, los procesos de acreditación se han constituido como un referente en nuestros días, ya que están destinados a garantizar la calidad, proporcionando credibilidad al proceso educativo y a sus resultados en beneficio de la instituciones y su comunidad [4].

La acreditación de un programa educativo posee una vigencia de cinco años y es otorgada por organismos acreditadores reconocidos por el COPAES, cuyo propósito es el de reconocer formalmente que el programa educativo cumple con criterios de calidad, fomentando la mejora continua a través de la atención de recomendaciones que la organización acreditadora emite a través de un dictamen derivado de la evaluación.

Derivado del seguimiento a las recomendaciones emitidas por los organismos CACECA y CONAIC a los programas educativos que oferta la Facultad de Negocios de la UNACH, cada cierto periodo de tiempo se realiza un proceso de recolección e integración de evidencias documentales de manera física y electrónica, los cuales se organizan por categorías e indicadores; en ese sentido, los Directivos y la Coordinación de Acreditación elaboran una planeación estratégica de acciones con la finalidad de atender las recomendaciones emitidas, por tal motivo, se solicita al personal docente actualicen el formato de Curriculum Vitae con información de productividad académica relativa a los últimos tres años consecutivos.

Por todo lo anterior, la principal problemática que se aborda en este trabajo académico, es la falta de automatización del proceso de captura de información en el formato de Curriculum Vitae que establece el organismo acreditador CONAIC [5]; el cual requiere ser llenado con información de índole laboral y de productividad académica del personal docente. En otras palabras, resulta ser un proceso laborioso mantener la información actualizada para atender cada proceso de evaluación de acreditación, lo cual se traduce en invertir tiempo en recolectar y clasificar la información de cada docente llenando el formato de Curriculum Vitae, y posteriormente generar reportes estadísticos; provocando en ocasiones inconsistencia en la información o que los formatos queden parcialmente llenados.

2.2. Marco teórico

2.2.1. Gestión de información

Hoy en día, el tratamiento de información representa un factor clave en las organizaciones, observándose un crecimiento exponencial de generación de datos de diversa índole, desde ese contexto las instituciones del mundo manejan grandes flujos de información haciendo resultado ser tedioso el proceso de extracción, transformación y análisis para coadyuvar a la toma de decisiones. Mientras más transcurre el tiempo los procesos de negocios se vuelven más complejos, el intercambio de información exige mayor celeridad y el número de bases de datos de información crece apresuradamente; por estas razones, la necesidad de los sistemas de gestión se convierte en primer orden para facilitar el manejo de la información, no solo ayuda en la toma de decisiones, sino también en la circulación interna de la información en los diferentes niveles de las organizaciones.

De acuerdo a lo manifestado por Suárez Alfonso, las organizaciones que manejan grandes flujos de información cuentan con sistemas de gestión para cumplir con sus objetivos. Cada organización debe precisar los recursos, las diferentes responsabilidades, metodologías, el tiempo de producción, y otros aspectos necesarios. Los procesos principales de la gestión de información son: la identificación, la adquisición de las fuentes informativas, su organización y almacenamiento, el desarrollo de productos y servicios, su distribución y uso [6].

2.2.2. Aplicaciones de Internet Enriquecidas / Rich Internet Application (RIA)

En los últimos años es evidente la evolución tecnológica y éxito que ha demostrado la Web 2.0. Esta evolución se observado especialmente en las interfaces de usuario, es evidente la creación de diversas tecnologías y bibliotecas de componentes que permiten mejorar el diseño de la interfaz. No hay que olvidar que la clave del éxito de la Web 2.0 es la participación del usuario final, es decir, la interacción del cliente con el sitio o aplicación web. Por consiguiente, es importante tener en claro que una aplicación web constituye un tipo especial de aplicación cliente/servidor, donde tanto el cliente como el servidor y el protocolo HTTP están estandarizados. Así también, es un conjunto de recursos web que participan simultáneamente para el funcionamiento de la misma, siendo estos: un servidor web, base de datos, bibliotecas de código y elementos web estáticos y dinámicos; sin embargo, este tipo de aplicaciones en sus inicios presentaban diversas problemáticas durante su ejecución, en otras palabras, eran poco interactivas, con una navegabilidad compleja, sucesivas recargas en el navegador web y un bajo performance en lo relativo a usabilidad y experiencia de usuario.

Como solución a los problemas antes mencionados, surge un paradigma tecnológico denominado Aplicaciones de Internet Enriquecidas o RIA por sus siglas en inglés. Este concepto fue introducido por diversos desarrolladores como Macromedia, enfatizando la necesidad que las aplicaciones Web tenían que ser más elaboradas, desde el punto de vista de la usabilidad de acuerdo a lo señalado por Mullet en su libro *La esencia de las aplicaciones ricas en Internet efectivas* [7]. Las aplicaciones desarrolladas bajo el paradigma RIA utilizan técnicas asíncronas como Ajax (JavaScript asíncrono) para permitir la comunicación de datos en tiempo real y la representación dinámica, lo que le da a la interfaz web una sensación de aplicación más similar a la de una aplicación de escritorio. De acuerdo a lo manifestado por Farrel Jason, el objetivo es simple: *dado que muchas personas viven en el escritorio, las aplicaciones de Internet adecuadas deben ser una extensión de donde viven; semejante a la zona de confort para trabajar correctamente* [8].

En este nuevo paradigma la característica principal es que el servidor ya no procesa el código relativo a la interfaz de usuario en virtud que dicho código reside o se ejecuta del lado del dispositivo cliente, en otras palabras, el navegador web. De tal suerte, que existe la posibilidad de responder a la interacción del usuario de forma inmediata al eliminar el retardo del esquema de comunicación petición-respuesta. Además, la información se proporciona de forma asíncrona, es decir, se da solución al problema del bloqueo de la interfaz de usuario. Por lo anteriormente expuesto, se seleccionó el framework Vue.js basado en el lenguaje JavaScript y la librería de componentes UI llamada PrimeVUE para la construcción del sistema de información SIACV abordado en este documento.

2.2.3. Interfaz de Programación de Aplicaciones/*Application Programming Interface* (API)

De acuerdo con lo manifestado en el sitio web de la empresa de tecnología RedHat, una API o Interfaz de Programación de Aplicaciones son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones; por esta razón, suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (petición) y el que requiere el productor (respuesta) [8].

Considerando lo anterior y desde otro contexto, el informático Roy Fielding fue el creador del concepto de transferencia de estado representacional (REST); en ese sentido, una API REST es un servicio web que utiliza la arquitectura REST para atender una solicitud realizada desde una aplicación cliente (*frontend*). En otras palabras, cuando el cliente envía una solicitud a través de una API de RESTful, esta transfiere una representación del estado del recurso requerido a quien lo haya solicitado. La información de respuesta se entrega por medio del protocolo HTTP en uno de estos formatos: JSON (JavaScript Object Notation), HTML, XLT o texto sin formato [9].

2.2.4. Patrón de diseño Modelo-Vista-Controlador (MVC)

Un patrón de diseño es básicamente una solución que surge de la experimentación práctica con varios proyectos de software. Cada patrón describe a un conjunto de objetos y clases comunicadas, este conjunto se ajusta para resolver un problema de diseño específico; posee cuatro elementos: nombre, problema, solución y consecuencias, que deben ser considerados al elegir el patrón de diseño más adecuado para desarrollar un proyecto.

En su sitio web educativo la Universidad de Alicante señala de manera específica la definición del patrón de diseño denominado Modelo-Vista-Controlador (MVC), como un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos [10]. Se trata de un modelo maduro que ha demostrado buenos resultados a lo largo de varios años al implementarse al proceso de desarrollo de diferentes tipos de aplicaciones, lenguajes y plataformas de desarrollo.

- El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia para el tratamiento de la información.
- La Vista, o interfaz de usuario prepara la información que se envía al cliente y los mecanismos interacción con los usuarios.
- El Controlador actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

2.2.5. Arquitectura Orientada a Servicios/*Service-Oriented Architecture (SOA)*

De acuerdo a lo manifestado por Erl Thomas, una arquitectura orientada a servicios (SOA) se debe entender como un conjunto de componentes, conectores, limitadores de composición, contenedores y configuraciones, que se unen de cierta manera para resolver problemas de arquitectura de software, los cuales a su vez intentan resolver problemas complejos de la realidad [11]. La arquitectura SOA separa los procesos de negocio de las funciones automatizadas y organiza estas últimas en módulos individuales catalogados en un diccionario de servicios que permiten su utilización por parte de toda la organización.

Un segundo aspecto que se debe considerar es el término "orientado a servicios", el cual se ha utilizado en diferentes contextos y propósitos. Una constante a lo largo de su existencia ha sido que representa un enfoque distinto para separar problemas. Lo que esto significa es que la lógica necesaria para resolver un gran problema se puede construir, ejecutar y gestionar mejor si se descompone en una colección de piezas relacionadas más pequeñas. Por esta razón en el proyecto que se aborda se utilizó la arquitectura orientada servicios.

3. Metodología

El desarrollo tradicional de un sistema de información difiere del desarrollo de aplicaciones o sistemas basados en un entorno web; los retos y los requerimientos para este tipo de aplicaciones difieren de un simple sitio web, resulta necesario pensar en diseñar una interfaz amigable, que brinde una correcta navegabilidad, usabilidad y experiencia de usuario; por estas razones es importante elegir una adecuada metodología de desarrollo de software.

3.1. Ingeniería Web (*IWeb*)

De acuerdo a Murugesan y Deshpande, la Ingeniería Web (*IWeb*) surge con el objetivo de aplicar los fundamentos de la Ingeniería del Software sobre el desarrollo sistemático de aplicaciones Web, atendiendo a las características particulares propias de este tipo de aplicaciones. Por lo tanto, se define como una disciplina específica, que estudia los procesos, métodos, técnicas y recursos esenciales para el desarrollo de aplicaciones Web de calidad [12].

Esta metodología consta de seis etapas que son ejecutadas bajo el contexto de un proceso incremental y evolutivo, lo que la convierte en un modelo eficiente para el desarrollo de sistemas web tal como se observa en la Figura 1.

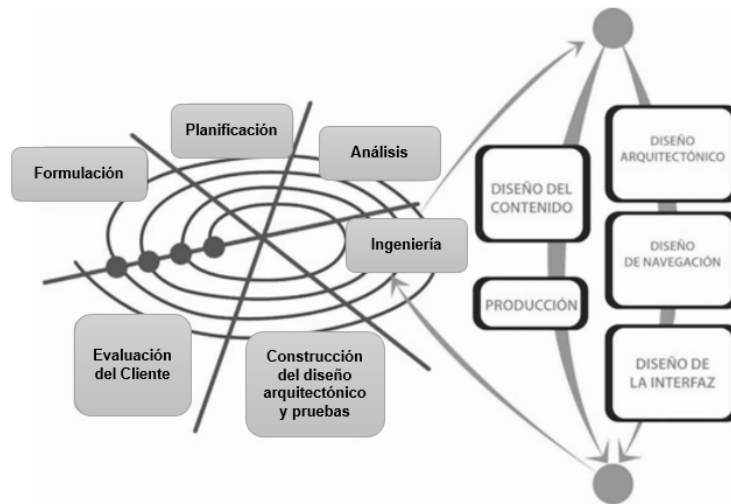


Figura 1. Metodología de software de Ingeniería Web (IWeb).

A continuación, se describen las principales acciones que se realizan en cada una de las etapas de la metodología IWeb con base lo señalado por Roger Pressman en su libro *Software Engineering. A Practitioner's Approach* [13].

1. **Formulación:** se identifican las metas y los objetivos del sistema, estableciendo de este modo la motivación del desarrollo del sistema, su importancia y los usuarios potenciales.
2. **Planificación:** se estima el costo global del proyecto y se evalúan los riesgos asociados con el esfuerzo del desarrollo, y se define una planificación del desarrollo muy detallada para el incremento final de la aplicación.
3. **Análisis:** se establecen los requisitos técnicos y de diseño, e identificación de los elementos de contenido que se van a incorporar. Durante esta etapa se realizan cuatro tipos de análisis diferentes: *Análisis del contenido*, *Análisis de la interacción*, *Análisis funcional* y *Análisis de la configuración*.
4. **Ingeniería:** se realizan las tareas diseño del contenido y producción, en paralelo con los diseños arquitectónicos, navegación e interfaz.
5. **Construcción:** se realiza la construcción del diseño arquitectónico haciendo uso de las herramientas para el desarrollo de aplicaciones web, sistemas y se asocia con el diseño arquitectónico, de navegación y de interfaz para la elaboración de web dinámicas.
6. **Pruebas:** se busca descubrir errores y ayuda a asegurar que la aplicación web funcionará correctamente en diferentes entornos. Para esto se hace uso de estrategias y técnicas que hayan sido recomendadas para otros sistemas.
7. **Evaluación del cliente:** se realizan todas las correcciones y cambios que se detectaron en la etapa de pruebas y se integran al sistema para el siguiente incremento, de tal modo que se asegure la satisfacción por parte del cliente, según los requerimientos solicitados.

3.2. Ejecución del modelo de Ingeniería Web (IWeb)

Con la finalidad de lograr un correcto desarrollo de la solución tecnológica planteada, se realizó la ejecución de cada una de las etapas que integran al modelo de proceso de Ingeniería Web (IWeb), destacando el valioso apoyo brindado por las autoridades y personal docente de la institución educativa.

1. **Formulación.** En esta primera etapa se identificaron los requerimientos para el desarrollo del sistema web, tales como: captura de los datos personales de cada docente, hacer uso del formato CVU que establece el organismo acreditador y la generación de reportes estadísticos; todo ello, proporcionado en diversas reuniones de trabajo por el personal directivo.

2. Planificación. Se elaboró una planificación del proyecto realizando una estimación de tiempos, estableciendo fechas de entrega de los avances; así también, se identificaron necesidades financieras y la evaluación de los posibles riesgos asociados al desarrollo del mismo.
3. Análisis. En esta etapa se realizó un análisis de requerimientos de acuerdo a la opinión de las partes interesadas, para comprender la problemática planteada y establecer los diseños de contenido que forman parte de la aplicación web; en virtud de ello se elaboraron diagramas de casos de uso (ver Figura 2), los cuales dan a conocer los actores, relaciones y principales acciones que pueden presentarse.

Los actores identificados son:

- *Coordinador*, es la persona responsable de administrar las cuentas y permisos de usuario. Así también, se encarga de validar los datos que fueron capturados usando el Sistema web.
- *Personal Docente*, es el docente que se encarga de capturar en el sistema web sus datos personales y de la producción académica generada en cierto periodo de tiempo; y generar su propio CV electrónico.
- *Personal Directivo*, es la persona que generan y analizan los reportes del sistema, para utilizarlos en la toma de decisiones.

Los casos de uso identificados son:

- Caso de uso crear nueva cuenta Usuario-Docente, permite registrar los datos básicos para crear una nueva cuenta de usuario de acceso al sistema.
- Caso de uso registrar datos, permite capturar los datos personales, grados académicos y de productividad del personal docente.
- Caso de uso consultar CVU, permite la consulta y generación automática de Curriculum Vitae del personal docente en forma electrónica (.pdf).
- Caso de uso reportes estadísticos permite generar diversos reportes estadísticos representados mediante gráficas que son generados con base a la información almacenada en la BD.

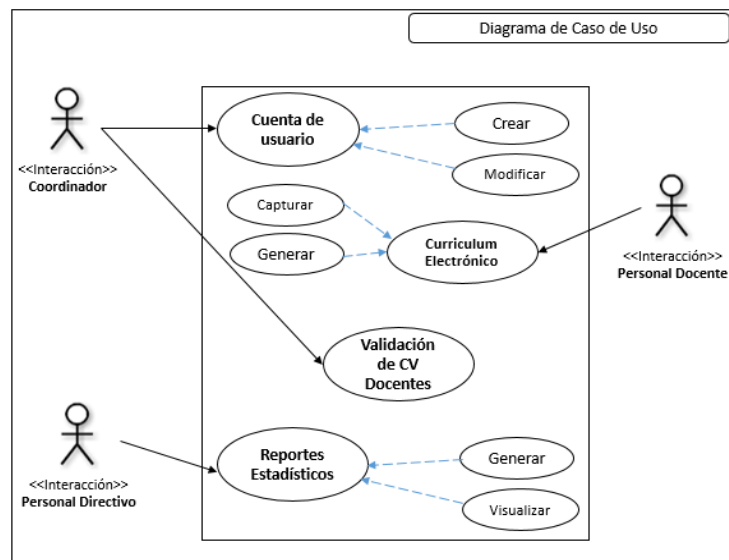


Figura 2. Diagrama de casos de uso diseñado con los principios de UML.

En una segunda fase se logró tener una buena comprensión de las características necesarias para realizar el diseño adecuado de la aplicación web; considerando que un buen diseño se enfoca a maximizar la simplicidad, consistencia, robustez y facilidad de uso del sistema. Posteriormente, se realizó un Diagrama Entidad/Relación de la base de datos, mismo que fue transformado a un modelo relacional, compuesto por varias tablas que almacenan datos personales, labores y de productividad académica del personal docente.

4. Ingeniería y construcción. En esta fase se realizaron un conjunto de actividades consistente en selección de *framework* y librerías de componentes UI, codificación, diseño de interfaz de usuario y refactorización. En el contexto de las aplicaciones web construidas con el *framework* Vue.js y la librería de componentes UI llamada PrimeVUE; todo ello bajo el patrón de diseño MVC y una arquitectura orientada a servicios.
5. Pruebas. Se efectuaron pruebas unitarias con la finalidad de encontrar errores de navegación, comunicación remota y seguridad en lo relativo a las sesiones de usuario.
6. Evaluación del cliente. En esta última etapa participaron el personal directivo y la coordinación de acreditación quienes son los principales usuarios del sistema de información, se realizó la evaluación en tiempo real de generar el formato electrónico CVU y los reportes solicitados en la etapa de formulación.

3.3. Implementación de la Arquitectura de Software

Un aspecto relevante en todo proyecto de software, es elegir la arquitectura idónea que permita cumplir con los objetivos del proyecto y lograr mayor calidad en el desarrollo; en consecuencia, para efectos de desarrollar el sistema de información propuesto, se eligió una arquitectura orientada a servicios dividiendo los requerimientos o la problemática a resolver, todo ello bajo un esquema de programación *backend* y *frontend*. De acuerdo con lo señalado por Marcano, una de las principales ventajas de implementar esta arquitectura es la fácil migración del proyecto en diferentes *frameworks* y tecnologías partiendo desde una lógica de negocios representada por servicios mediante una comunicación remota con un API REST [14].

Con la interacción remota con los servicios se busca separar la lógica de negocio de tal suerte que coadyuve a la escalabilidad y mantenimiento del software; al mismo tiempo establecer un flujo claro en la información con la finalidad de identificar posibles errores que surjan en la etapa de producción del proyecto, efectuándose correcciones o mejoras en los componentes de la lógica de negocio sin necesidad sin afectar el funcionamiento del resto de los módulos; por lo anterior, se realizó una programación de lado del servidor con lenguajes de programación específicos que permiten la generación de una solución tecnológica orientada a servicios.

3.3.1. Programación del lado del Servidor (*Backend*)

Para efectuar de manera correcta la programación del lado del servidor, se utilizó el *framework* de código abierto denominado Codeigniter que ofrece una alta robustez para crear software que se ejecuta de lado servidor, el cual está basado en lenguaje PHP, y posee ciertas bibliotecas que permiten la creación de componentes que atienden a las múltiples peticiones que llegan mediante el protocolo HTTP [15]. Además, ofrece mecanismos de seguridad basados en técnicas de cifrado como lo es JSON Web Token (JWT), realizando un filtrado de las peticiones dando respuesta únicamente a las emitidas por usuarios autorizados. En la Figura 3 se muestra un esquema de la programación *backend* utilizada.

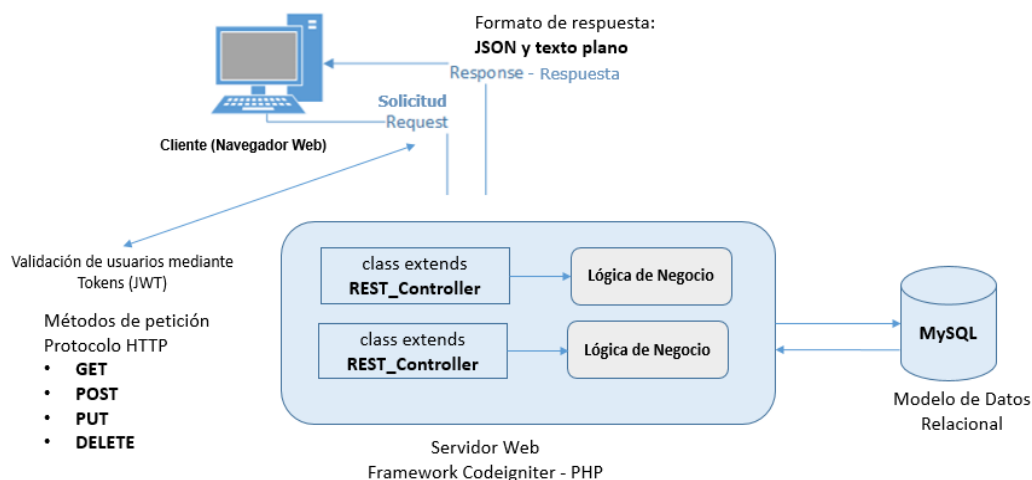


Figura 3. Arquitectura orientada a servicios bajo el contexto de una programación *backend*. Fuente: Elaboración propia.

En la Figura 3 es posible observar un esquema de trabajo que posee tres componentes fundamentales, que permiten el correcto funcionamiento de una API REST desde la perspectiva de una programación de lado del servidor.

- El Controlador es el componente que contiene los métodos que representan los recursos que se pueden consumir por el cliente, en otras palabras, son métodos que responden a las peticiones realizadas a través del protocolo HTTP.
- La capa de Lógica de negocio posee métodos específicos que almacenan ciertas reglas o restricciones de funcionamiento, es decir, se encargan de realizar tareas específicas para dar solución a una problemática. Además, son el puente de comunicación hacia la capa de datos para recuperar la información necesaria.
- La capa de Datos posee los métodos de conexión hacia el modelo de datos relacional; así también hace referencia a ciertos procedimientos almacenados en el gestor de la base de datos. En esta capa se agrupan las sentencias para efectuar las operaciones básicas de manipulación de datos.

3.3.2. Programación del lado del Cliente (*frontend*)

En lo que respecta a la programación del lado del cliente o desarrollo *frontend* se utilizó el *framework* JavaScript de código abierto denominado Vue.JS creado por Evan You liberado bajo la licencia MIT [16]. Esta tecnología se centra en el diseño de la interfaz bajo la perspectiva de funcionamiento de una página. La ventaja principal de esta técnica de programación es que permite modificar cada una de estas partes sin tener que modificar las demás, cada una de las interfaces de usuario son modulares y escalables. Para diseñar las interfaces de usuario se utilizó la biblioteca de componentes UI de código abierto llamada PrimeVUE creada por la empresa PrimeTek, la cual es una potente biblioteca versátil y de alto rendimiento que ofrece más de 80 componentes de interfaces de usuario y con la opción de instalar plantillas profesionales que permiten cambiar el aspecto visual en lo que respecta a colores, iconos y tipografía; además la mayor parte de estas plantillas ofrecen un diseño responsivo permitiendo que la interfaz de usuario pueda ser desplegada en cualquier dispositivo móvil que posea un navegador web.

Otro aspecto relevante relacionado con el funcionamiento de la Interfaz de usuario, es la generación de peticiones remotas mediante HTTP para consumir los recursos que ofrece la API REST, para ello se utilizó la biblioteca JavaScript llamada Axios, la cual es compatible con el *framework* Vue.js y que permitió la generación de peticiones usando los métodos de comunicación asíncrona POST, GET, PUT, DELETE desde el lado del Cliente; cabe señalar que Axios recibe los datos de respuesta en formato JSON, los procesa y posteriormente los carga en estructuras de datos dinámicas para ser manipulados con Vue.js y visualizarlos a través de los componentes de interfaz de usuario que ofrece PrimeVUE, teniendo como un claro ejemplo el uso de un DataTable. En la Figura 4 se puede observar una representación gráfica del funcionamiento de las tecnologías señaladas.

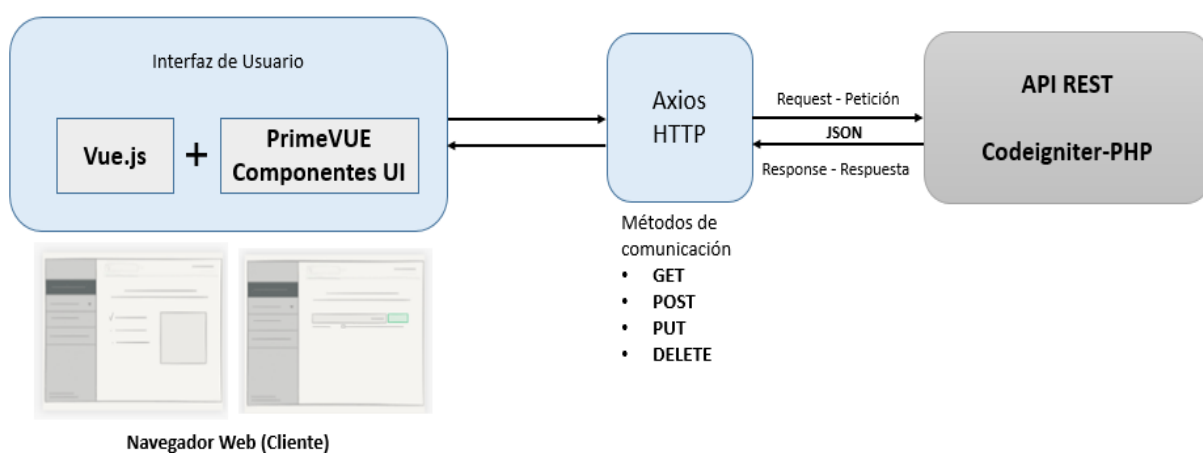


Figura 4. Programación del lado del cliente usando tecnologías basadas en JavaScript.

Fuente: Elaboración propia.

3.4. Diseño del Modelo de Datos Relacional

En lo que respecta al almacenamiento de los datos se diseñó e implementó una base de datos considerando los principios y reglas del modelo relacional creado en el año de 1970 por Edgar Frank Codd, por ende, se realizó la creación de tablas y se establecieron relaciones entre las mismas; este tipo de bases de datos garantiza la integridad referencial, es decir, al eliminar un registro se eliminan todos los registros relacionados dependientes. Así también, favorece la normalización por ser más comprensible y aplicable. Considerando lo mencionado, se seleccionó el sistema gestor de bases de datos multiusuario, relacional y multihilo denominado MySQL, reconocido como uno de los SGBD de código abierto más populares del mundo, otra característica relevante es su alta compatibilidad con entornos de desarrollo para aplicaciones web destacando su eficiente tiempo de respuesta en la ejecución de consultas de datos de manera remota. En la Figura 5 se puede observar el diagrama entidad-relación utilizado como fuente para crear el modelo de datos relacional utilizado en el desarrollo del proyecto Sistema de Información Web SIACV versión 1.0.

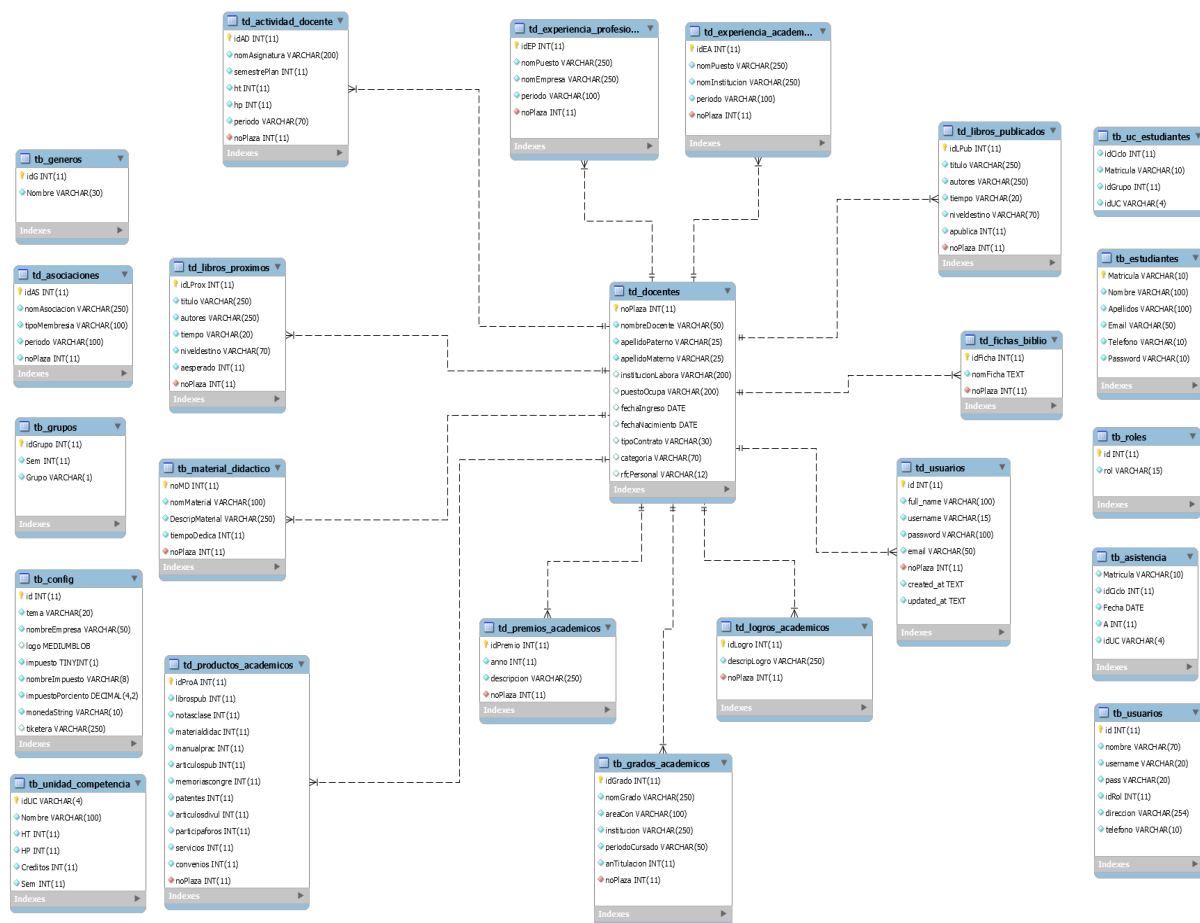


Figura 5. Diagrama Entidad-Relación del Sistema de Información Web SIACV v1.0

Fuente: Elaboración propia.

4. Resultados

La solución tecnológica desarrollada Sistema de Información Web SIACV, se considera una innovación en el ámbito de la gestión de la información dentro de la Institución Educativa, en virtud de la arquitectura de software, metodología y *framework* para el desarrollo del mismo; así también, se considera un trabajo académico colaborativo sumamente gratificante debido que la propuesta se derivó de una necesidad real de la Institución Educativa. Así también, es importante especificar que el tipo de arquitectura utilizada facilita la interacción con servicios web remotos permitiendo separar la lógica de negocio y de la interfaz gráfica de tal forma que facilita la

posibilidad de un crecimiento futuro, tal como lo manifiesta Luna Santos [17]. Cabe señalar que este proyecto representa una transferencia de tecnología beneficiando de manera directa al personal directivo y docente.

Por otra parte, hay que resaltar que el proceso de liberación de los diferentes módulos o componentes que integran al sistema de información, se realizó de forma paulatina e incremental, siendo ejecutados en un servidor de pruebas y posteriormente instalados en un servidor web de producción, que se encuentra asociado al dominio institucional (<http://negocios.unach.mx/siacv/>) de la Facultad de Negocios de la UNACH.

Para efectos de concretar la entrega del proyecto a los directivos de la institución, se realizó la creación de cuentas de usuario con permisos específicos tanto de nivel administrador para los directivos y cuentas personales para cada docente, mismas que son necesarias para validar su ingreso al sistema de información web, el cual genera un *token* de sesión para cada usuario activo, para ello, se implementó la tecnología de JWT (Json Web Token) [18].

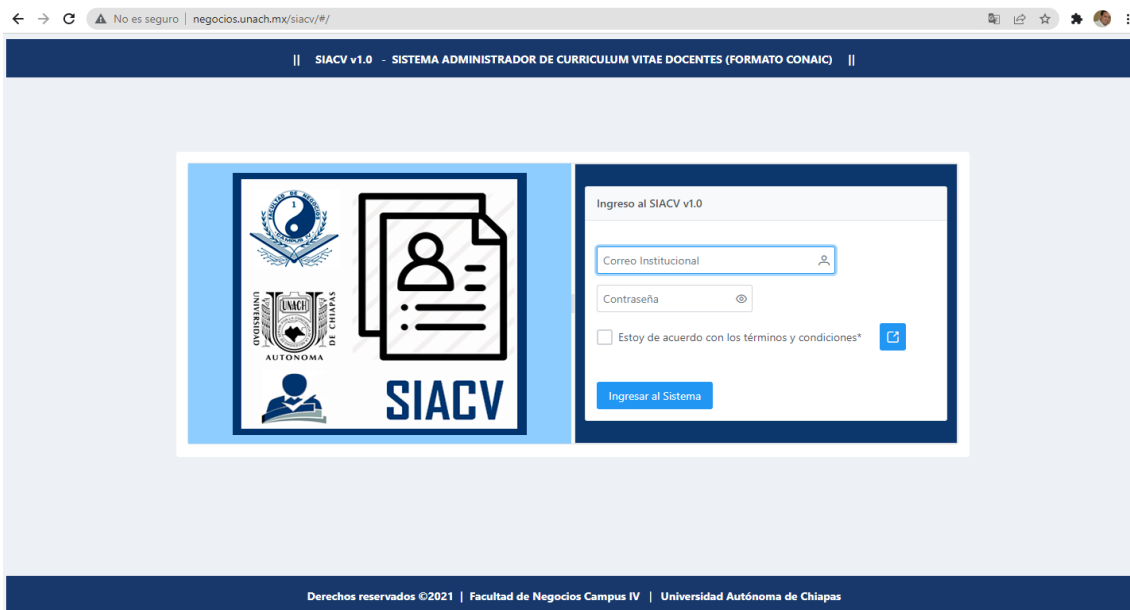


Figura 6. Pantalla de Inicio de Sesión.

Fuente: Elaboración Propia.

En la Figura 6 se observa la interfaz de usuario para el inicio de sesión al sistema, lo que significa que el usuario debe ingresar su correo institucional (cuenta de usuario) y la contraseña respectiva, así también debe indicar que está de acuerdo con los términos y condiciones. Una vez efectuada la validación del usuario, se carga la interfaz del menú principal o *Dashboard*, la cual permite visualizar en la parte superior derecha el nombre y número de plaza del docente; en el lado izquierdo se despliega un menú de navegación que ofrece opciones organizadas por categorías y/o módulos; siendo la categoría principal CONAIC tal como se visualiza en la Figura 7.

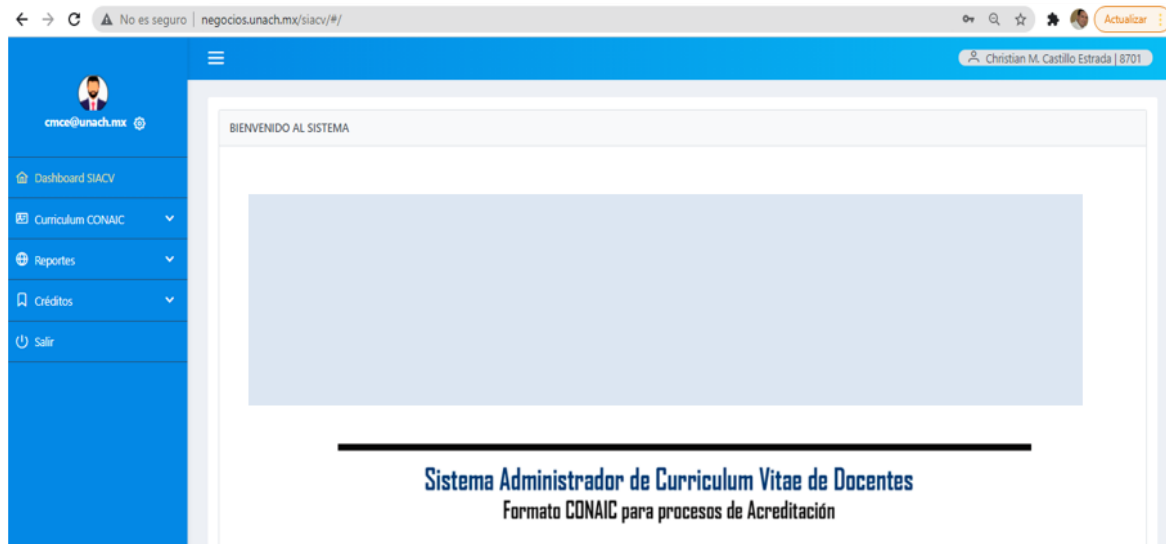


Figura 7. Pantalla del Menú Principal.
Fuente: Elaboración Propia.

Figura 8. Pantalla de Registro de Datos Personales.
Fuente: Elaboración Propia.

La primera opción del menú dentro de la categoría Curriculum CONAIC se denomina Datos Personales, tal como se observa en la Figura 8; la cual despliega una interfaz de usuario que permite visualizar y actualizar los datos del docente, es decir, nombre completo, puesto, adscripción, número de plaza, tipo de contrato, categoría, fecha de nacimiento y fecha de ingreso a la institución.

Así también, se visualiza una segunda opción que corresponde a los Grados Académicos que posee el docente, para tal efecto se implementó un componente UI denominado *DataTable* que despliega todos los registros almacenados en la base de datos, y por cada uno de ellos, existen los botones de acción de editar y eliminar ubicados en la última columna como se muestra en la Figura 9. Además, cuando el usuario da clic en el botón registrar se visualiza un formulario para la captura de un nuevo registro de grado académico.

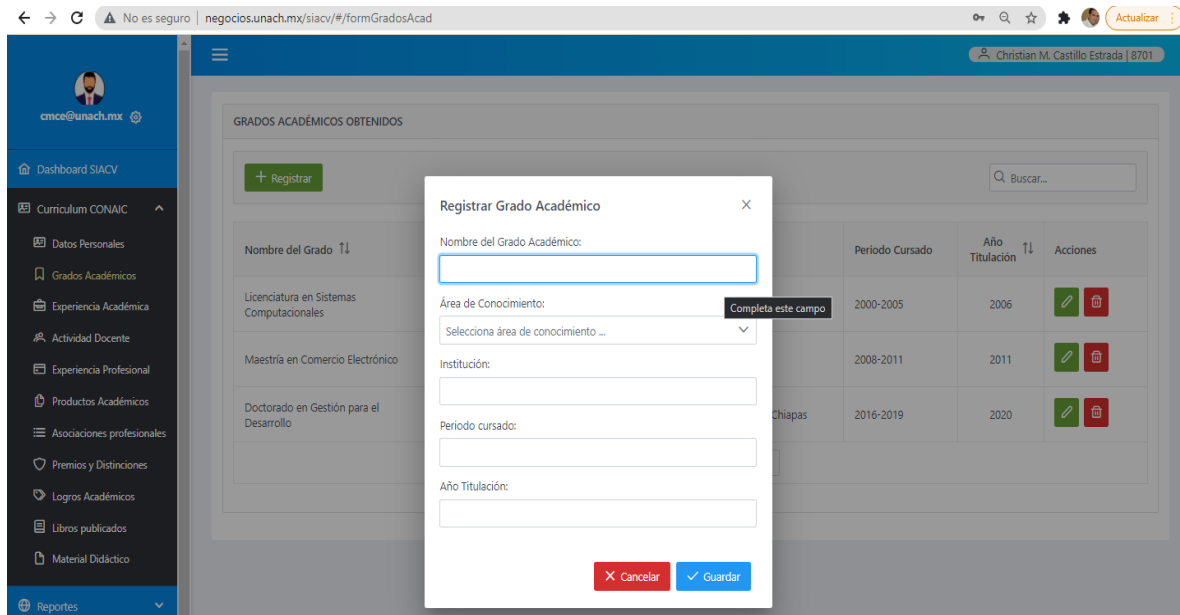


Figura 9. Interfaz de captura de grados académicos.
Fuente: Elaboración Propia.

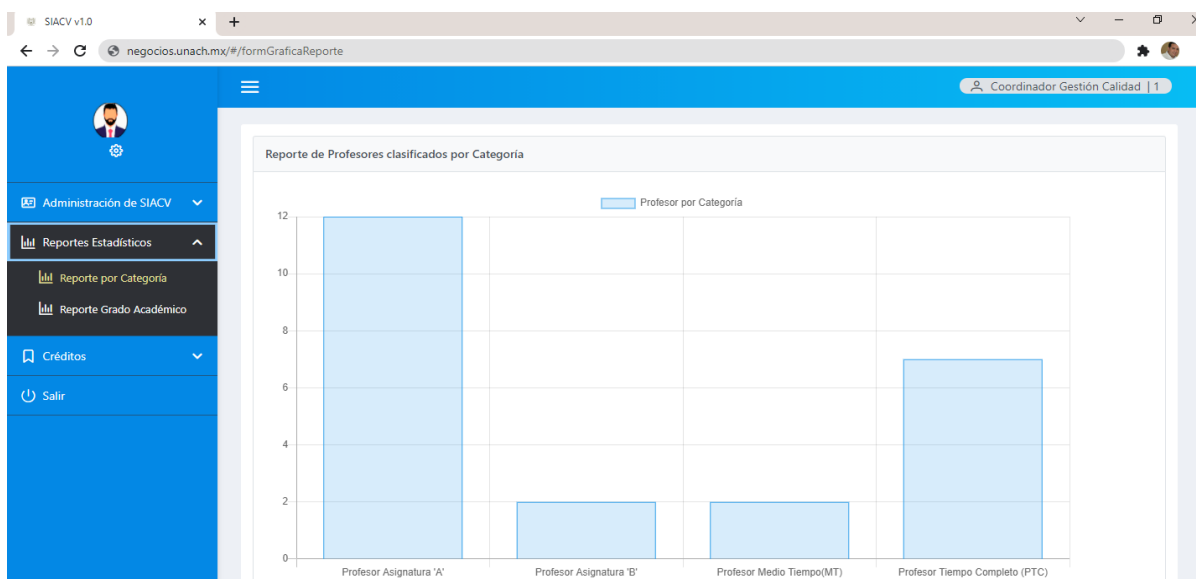


Figura 10. Reporte estadísticos personalizados.
Fuente: Elaboración Propia.

Con la finalidad de solventar el requerimiento de generar evidencias con datos estadísticos que solicita el organismo acreditador, se desarrolló un módulo relativo a la generación de Reportes Estadísticos con base a la información de datos personales del personal docente adscrito a la Institución Educativa. En la Figura 10 se muestra una gráfica de barras la cual se despliega en tiempo real con base a los datos consultados y obtenidos de manera remota consumiendo los recursos del API REST. Lo anterior, me permite corroborar lo manifestado por Ana Sifuentes, quien señala que los sistemas en línea brindan resultados inmediatos y permiten establecer medios de evaluación homogéneos, con capacidad de adaptación al cambiante mundo globalizado [19].

Por otra en la Figura 11 se observa el formato de Curriculum Vitae Electrónico que establece el organismo acreditador, este módulo genera un documento en formato PDF automáticamente, mismo que puede ser descargado en un equipo de cómputo o bien se puede imprimir directamente; la información que contiene el formato corresponde a todos los datos que capturó el docente en cada uno de los módulos y que se encuentra almacenada en el sistema gestor de base de datos.

Formato CONAIC de Currículum Vitae Docente

print_cvdocente_pdf.php 1 / 2 90%

CURRÍCULUM VITAE RESUMIDO - FORMATO CONAIC
(No incluir anexos ni comprobantes)

APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE(S)
Castillo	Estrada	Christian Mauricio

INSTITUCIÓN EN QUE LABORA ACTUALMENTE	PUESTO EN LA INSTITUCIÓN	FECHA DE INGRESO
Facultad de Negocios Campus IV de la UNACH	Profesor de Tiempo Completo	2006-07-07

TIPO DE CONTRATO	NO. PLAZA/PROFESOR	FECHA DE NACIMIENTO
Definitivo	8701	1983-02-22

NOMBRE DEL GRADO ACADÉMICO Y ÁREA	INSTITUCIÓN	PERIODO	AÑO TIT.
Licenciatura en Sistemas Computacionales	UNACH	2000-2005	2006
Maestría en Comercio Electrónico	ITESM	2008-2011	2011
Doctorado en Gestión para el Desarrollo	Universidad Autónoma de Chiapas	2016-2019	2020

ACTIVIDAD O PUESTO (EXPERIENCIA ACADÉMICA)	ORGANIZACIÓN O INSTITUCIÓN	AÑO PERIODO
Coordinador de Planeación	Facultad de Contaduría Pública CIV UNACH	02/2008 a 05/2015
Profesor de Asignatura "A"	Facultad de Contaduría Pública CIV UNACH	01/2007 a 11/2014
Secretario Académico	Facultad de Negocios Campus IV UNACH	05/2015 a 05/2019
Profesor de Tiempo Completo Titular A	Facultad de Negocios Campus IV UNACH	12/2014 a la Fecha

Figura 11. Generación del CVU del Docente.
Fuente: Elaboración Propia.

5. Conclusiones

Un aspecto clave posterior a la implementación y pruebas del sistema de información web, es la disminución en los tiempos de organización de evidencias documentales y captura de información por parte del personal docente. Así también, se observa la generación rápida de reportes estadísticos que coadyuvarán durante las evaluaciones para la acreditación de los programas educativos que oferta la institución educativa, resultando un beneficio para los directivos y la comunidad universitaria al contar con información precisa y confiable.

En otro orden de ideas, la utilización de una arquitectura de software basada en servicios permitió facilitar el desarrollo e implementación de la aplicación web; y se generó una base sólida para el crecimiento o mantenimiento futuro del sistema, en virtud de que dicha arquitectura separa la lógica de negocio y de datos por un lado, es decir la programación de lado del servidor; y por otra parte todo lo relacionado a la programación de la interfaz gráfica; sin generar problemas de compatibilidad o desempeño. Aunado a ello, resulta importante señalar que fue utilizada tecnología exclusivamente de código abierto, la cual no generó problemas durante las etapas de desarrollo.

Sin duda, este proyecto tecnológico representó para los estudiantes que participaron, una experiencia académica de gran beneficio para su formación profesional, generando en ellos nuevas competencias profesionales, como fue el uso y aplicación de una metodología de desarrollo de software, aprendizaje de nuevos lenguajes de programación de última generación.

Como valor agregado del proyecto en comparación a otras plataformas o sistemas, se destaca la gran escalabilidad proporcionada por la arquitectura orientada a servicios y las tecnologías utilizadas; aunado a que su implementación se encuentra sobre una plataforma en la nube y servicios dedicados de internet administrados por la Universidad que garantizan el funcionamiento, estabilidad y disponibilidad de la información. El área de oportunidad que ofrece es la construcción de nuevos módulos o el desarrollo de una aplicación móvil que consuma el mismo servicio, ya que la lógica de negocio que poseen los diferentes recursos del API REST no requiere de mayores adecuaciones.

6. Agradecimientos

Se expresa un cordial agradecimiento a todo el personal directivo y docente de la Facultad de Negocios de la Universidad Autónoma de Chiapas, por el valioso apoyo y las facilidades que brindaron para la realización de este proyecto; así también, un valioso reconocimiento a los estudiantes que participaron en las etapas del proyecto.

7. Referencias

- [1] CACEI. (2021). *¿Qué es un Programa Acreditado?* Recuperado de: <http://cacei.org.mx/nvtu/nvtu01/nvtu0101.php>
- [2] COPAES. (2021). *Origen de COPAES*. Recuperado de: <https://www.copaes.org/copaes.html>
- [3] Rueda, M. (2009). Conferencia Regional de la Educación Superior en América Latina y el Caribe. Declaraciones y plan de acción. *Perfiles Educativos*, 31 (125), 90-108. Recuperado de: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0185-26982009000300007
- [4] Castillo Marrufo, J. A., Hernández Jaime, J., Aragón García, M. (2014). Los procesos de acreditación: desafíos para la Educación Superior en México. *Revista Iberoamericana de Producción Académica y Gestión Educativa*, 1 (1), 1-9. Recuperado de: <https://www.pag.org.mx/index.php/PAG/article/view/118>
- [5] CONAIC. (2020). *Formato de Curriculum Vitae Resumido (CVU)*. Recuperado de: <https://www.conaic.net/formatos.html>
- [6] Suárez Alfonso, A., Cruz Rodríguez, I., Pérez Macías, Y. (2015). La gestión de la información: Herramienta esencial para el desarrollo de habilidades en la comunidad estudiantil universitaria. *Revista Universidad y Sociedad*, 7 (3), 72-79. Recuperado de: <http://scielo.sld.cu/pdf/rus/v7n2/rus10215.pdf>
- [7] Mullet, K. (2003). *The Essence of Effective Rich Internet Applications. Macromedia White Paper*. Recuperado de: https://darmano.typepad.com/for_blog/essence_of_ria.pdf
- [8] Red Hat. (2020). *API o interfaz de programación de aplicaciones ¿Qué es una API de REST?* Recuperado de: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [9] IBM. (2021). *REST APIs Design*. Recuperado de: <https://www.ibm.com/cloud/learn/rest-apis>
- [10] Universidad de Alicante. (2021). *Modelo vista controlador (MVC)*. Recuperado de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- [11] Erl, T. (2005). *Service-Oriented Architecture Concepts, Technology, and Design*. USA: Prentice Hall.
- [12] Murugesan, S., Deshpande, Y., Hansen, S., Ginige, A. (2001). Web Engineering: a New Discipline for Development of Web-Based Systems. En S. Murugesan, Y. Deshpande (Eds.), *Web Engineering. Lecture Notes in Computer Science* (pp. 3-13). Berlin: Springer. doi: https://doi.org/10.1007/3-540-45144-7_2
- [13] Pressman, R. (2010). *Software engineering. A Practitioner's Approach*. Madrid: McGraw-Hill.
- [14] Marcano, A. (2020). *ASP.NET CORE 3.1 MVC Y WEB API: Ejemplos y Ejercicios en C#* (Spanish Edition). Independently published.
- [15] CodeIgniter Oficial. (2020). *Guía del Usuario*. Recuperado de: <https://codeigniter.es/documentacion.php>
- [16] Vue.js. (2021). *Guía de Introducción a Vue.js*. Recuperado de: <https://es.vuejs.org/v2/guide/>
- [17] Luna Santos, R., Santos Quiroz, R. A., Carrera Carrasco, E., González Domínguez, I., Santaella Benavente, M. G. (2021). Creación e implementación de una aplicación web para la administración de citas para un autolavado. *Revista de Investigación en Tecnologías de la Información (RITI)*, 9 (19), 97-108. doi: <https://doi.org/10.36825/RITI.09.19.008>
- [18] López Magaña, L. M. (2020). *Qué es Json Web Token y cómo funciona*. Recuperado de: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- [19] Sifuentes Ocegueda, A. T. (2019). Sistema web para la evaluación 360° de docentes de nivel superior de la Universidad Autónoma de Nayarit, México. *Revista de Investigación en Tecnologías de la Información (RITI)*, 7 (14) 215-223, nov. 2019. Recuperado de: <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/217>