

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



PROTOTIPO DE SISTEMA DE EMISIÓN DE CONSTANCIAS
ELECTRÓNICAS DE ESTUDIANTE INSCRITO PARA EL IIT
DE LA UACJ

Reporte Técnico de Investigación presentado por:

César Alonso Rivas Flores 143468

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Mtro. René Noriega Armendáriz

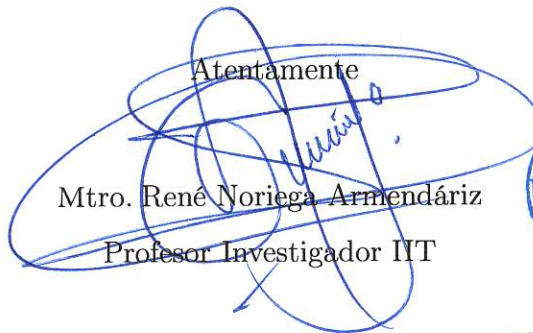
Ciudad Juárez, Chihuahua, a 6 de Noviembre de 2019

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente me permito comunicarle que, después de haber realizado las asesorías correspondientes al reporte técnico Prototipo de sistema de emisión de constancias electrónicas de estudiante inscrito para el IIT de la UACJ, del alumno César Alonso Rivas Flores, de la Licenciatura en Ingeniería en Sistemas Computacionales, considero que lo ha concluido satisfactoriamente, por lo que puede continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente

Mtro. René Noriega Armendáriz
Profesor Investigador IIT



Ccp:
Coordinador del Programa de Sistemas Computacionales
César Alonso Rivas Flores
Archivo

Ciudad Juárez, Chihuahua, a 6 de Noviembre de 2019

Asunto: Autorización de Publicación

C. César Alonso Rivas Flores

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la publicación del documento de Prototipo de sistema de emisión de constancias electrónicas de estudiante inscrito para el IIT de la UACJ, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.



Mtra. Ivonne Haydee Robledo Portillo

Profesor Titular de Seminario de Titulación II

Declaración de Originalidad

Yo, César Alonso Rivas Flores, declaro que el material contenido en esta publicación fue elaborado con la revisión de los documentos que se mencionan en el capítulo de Bibliografía, y que la solución obtenida es original y no ha sido copiada de ninguna otra fuente, ni ha sido usada para obtener otro título o reconocimiento en otra institución de educación superior.



César Alonso Rivas Flores

Agradecimientos

Expreso mi más sincero agradecimiento a todos mis maestros, que a lo largo de esta aventura supieron guiarme en cada paso, además de darme esa gran motivación y empuje para tener siempre una actitud positiva, especialmente a mi maestro, asesor, *sensei*, compañero y amigo el Mtro. René Noriega Armendáriz, a mi maestra Ivonne Haydee Robledo Portillo y mi profesor Roberto Fierro Balderrama, a quienes admiro y agradezco por tantos buenos momentos, consejos y aliento, al maestro Natividad Nieto Saldaña, por todo su apoyo a lo largo de estos años, al Dr. Juan Francisco Hernández Paz y a la Dra. Alejandra Rodríguez, por la motivación y el apoyo para desarrollarme profesionalmente y cultivar mis valores, al Mtro. Mario Silvino Ávila, Ángeles Arellano, Erika Castorena y Nancy Nevárez, por todo su apoyo y amistad, a la Universidad Autónoma de Ciudad Juárez, por convertirse en mi alma mater.

Dedicatoria

Este trabajo de investigación, está dedicado a mi padre Jesús Renato Rivas Esparza, quién vivió su vida dándome el ejemplo de una buena persona, un abrazo hasta el cielo papá. A mi madre Rosa María Flores Castañeda, por todo su esfuerzo, por estar al pie del cañón en todo momento, te amo mamá, a mis hijos Renato, César y Alonso, por ser el motor que impulsa cada paso que doy, a mi tía Gregoria Aguilera Sánchez, por su ayuda y apoyo en los momentos difíciles, a mis hermanos René, Roxana y Diana y a mi sobrino Eliu Calzada, por su apoyo y sus palabras que me hicieron sentir orgulloso. Gracias Dios por bendecirme y darme infinitas fuerzas a través de tantas personas, para culminar mi carrera profesional.

Índice general

1. Planteamiento del Problema	2
1.1. Antecedentes	2
1.2. Definición del problema	4
1.2.1. Problemas identificados	4
1.3. Objetivos	6
1.3.1. Objetivo general	6
1.3.2. Objetivos específicos	6
1.4. Justificación	6
2. Marco teórico	8
2.1. Metodologías de análisis, modelado de procesos y aplicaciones	8
2.1.1. Definición de metodología	8
2.1.2. Herramientas de modelado de procesos	9
2.1.3. Herramientas de modelado de aplicaciones	12
2.2. Bases de datos	16
2.2.1. Definición de base de datos	18

2.2.2.	Clasificación de bases de datos	18
2.2.3.	Modelos de bases de datos	19
2.2.4.	Sistemas de gestión de bases de datos (SGDB)	20
2.3.	Lenguajes de programación y herramientas de diseño de aplicaciones web . .	21
2.3.1.	Definición de lenguaje de programación	21
2.3.2.	PHP	22
2.3.3.	NodeJS	22
2.3.4.	Javascript	22
2.3.5.	HTML	22
2.3.6.	CSS	23
2.3.7.	MVC	23
2.3.8.	<i>Bootstrap</i>	23
2.4.	Metodologías de desarrollo de software	24
2.4.1.	Definición de metodología	24
2.4.2.	Ciclo de vida del software	24
2.4.3.	Metodologías de desarrollo tradicionales	25
2.4.4.	Metodologías de desarrollo ágiles	26
2.5.	Técnicas e instrumentos de recolección de datos	27
2.5.1.	Cuestionario	30
2.5.2.	Validación de instrumento	30
2.5.3.	Confiabilidad de instrumento	30

2.6. Seguridad de la información	30
2.6.1. Concepción de la seguridad de la información	31
2.6.2. Planificación de la seguridad	32
2.6.3. Tecnologías	32
2.6.4. Estándares de seguridad de la información	33
2.7. Usabilidad	34
2.7.1. Conceptos sobre usabilidad	34
2.7.2. Estándares de usabilidad	35
2.7.3. Guías de estilo	36
2.7.4. Atributos de la usabilidad	36
2.7.5. Evaluación de la usabilidad	37
3. Desarrollo del Proyecto	39
3.1. Producto propuesto	39
3.2. Descripción de la metodología	41
3.2.1. Análisis del proceso actual de emisión de constancias	42
3.2.2. Determinar los requerimientos del sistema	46
3.2.3. Modelado de las bases de datos del sistema	50
3.2.4. Diseño del prototipo del sistema	57
3.2.5. Desarrollo del prototipo del sistema	58
4. Resultados y Discusiones	65
4.1. Presentación de resultados	65

<i>ÍNDICE GENERAL</i>	x
4.2. Interpretación de resultados	69
5. Conclusiones	71
5.1. Con respecto al objetivo de la investigación	71
5.2. Recomendaciones a futuro	72
Bibliografía	73
A. Formato de solicitud de constancias departamento de física y matemáticas	79
B. Estructura del proyecto	80
C. Funciones principales	81

Índice de figuras

2.1. Diagrama de casos de uso	11
2.2. Diagrama BPMN	12
2.3. Diagrama de estados	13
2.4. Símbolos básicos de los DFD	13
2.5. Diagrama de clases	15
2.6. Modelo de dominio	16
2.7. Diagrama de secuencia	17
2.8. Iteración RUP	27
2.9. Variación de esfuerzo de acuerdo a la fase del proyecto	28
2.10. Esquema de proceso Scrum	29
2.11. Coeficiente KR20	31
3.1. Diagrama conceptual del prototipo de sistema de emisión de constancias de estudiante inscrito	40
3.2. BPMN proceso actual departamento física y matemáticas	43
3.3. BPMN proceso actual departamento ingeniería eléctrica y computación . . .	44

3.4. BPMN proceso actual departamento ingeniería industrial y manufactura . .	45
3.5. BPMN proceso actual departamento ingeniería civil y ambiental	46
3.6. Cantidad de créditos por semestre del programa de ingeniería en sistemas computacionales	49
3.7. Esquema de base de datos del prototipo de sistema de emisión de constancias de estudiante inscrito	51
3.8. Modelo de dominio, módulo de coordinadores	53
3.9. Modelo de dominio, módulo de constancias de estudiante inscrito	54
3.10. Modelo de dominio, módulo de validación de constancias de estudiante inscrito	55
3.11. Diagrama de casos de uso para el módulo de coordinadores	56
3.12. Diagrama de casos de uso para el módulo de constancias de estudiante inscrito	56
3.13. Diagrama de casos de uso para el módulo de validación de constancias de estudiante inscrito	57
3.14. Prototipo de pantalla de inicio de sesión	58
3.15. Diagrama de navegación entre módulos del prototipo de sistema	59
3.16. Función de redireccionamiento de vistas, incluida en controladores de módulos	60
3.17. Función de validación de constancia, incluida en el controlador del módulo de validaciones	61
3.18. Componentes para el módulo de coordinadores	62
3.19. Función para la generación de constancia digital en PDF	62
3.20. Prototipo de constancia electrónica en formato PDF	64
4.1. Pantalla principal del módulo de coordinadores	66

4.2. Pantalla principal módulo de estudiante	67
4.3. Pantalla de resultado de validación de constancia electrónica	67
4.4. Porcentaje de funcionamiento módulo de estudiantes	68
4.5. Tasa de finalización de tareas	70
4.6. Índice de satisfacción general	70
A.1. Formato de solicitud de constancias departamento de física y matemáticas. .	79
B.1. Estructura de directorios del sistema.	80
C.1. Función para actualizar la tabla de apoyo Siiv2.	81
C.2. Función principal del módulo de administrador.	82
C.3. Función para extraer de la B.D. la firma de un coordinador.	83
C.4. Función para extraer de la B.D. el sello de una coordinación.	83
C.5. Función principal del módulo de constancias de estudiante inscrito.	84
C.6. Función para consultar las constancias generadas por un estudiante.	85
C.7. Controlador de inicio de sesión en el sistema.	86
C.8. Función principal del módulo de coordinadores.	87
C.9. Función para eliminar registros de un determinado programa educativo. . . .	87
C.10.Función para la generación de la constancia digital en formato PDF.	88
C.11.Función para la generación de la constancia digital.	89
C.12.Función extraer de la B.D. los programas educativos de un coordinador en específico.	89
C.13.Middleware para el módulo de administrador.	90

C.14.Middleware para el módulo de estudiantes.	91
C.15.Middleware para el módulo de coordinadores.	92
C.16.Función para la reimpresión de constancias.	93
C.17.Función para calcular el semestre en el que un estudiante se encuentre inscrito con base en los créditos acreditados y cursados.	94
C.18.Función para actualizar en la B.D. la firma digitalizada del coordinador. . . .	94
C.19.Función para actualizar en la B.D. el sello digitalizado de las coordinaciones.	95
C.20.Función para validar una constancia generada por el sistema.	95
C.21.Función principal del módulo de validación.	96

Índice de tablas

3.1. Porcentaje de ocurrencia de elementos en constancias analizadas.	47
3.2. Dependencias principales, identificadas en el análisis de requerimientos. . . .	48

Resumen

Cualquier desarrollo tecnológico enfocado a una adecuada gestión de la información, trae grandes beneficios para las organizaciones e instituciones que apuestan a ellos, esta investigación tuvo como objetivo la automatización del proceso de emisión de constancias de estudiante inscrito en el Instituto de Ingeniería y Tecnología (IIT) de la Universidad Autónoma de Ciudad Juárez (UACJ). Para la cual, se realizaron entrevistas directas a 10 coordinadores de programas educativos, así como a 9 secretarías de los diferentes departamentos del IIT, además, se analizaron un total de 50 constancias de estudiante inscrito. Para el desarrollo del prototipo de sistema, se empleó la metodología RUP (Proceso Unificado Racional), donde se realizaron 8 casos de uso, 6 diagramas BPMN para el modelado de los procesos, 3 modelos de dominio y diagrama de navegación, para en conjunto, establecer los requerimientos del prototipo de sistema de constancias electrónicas, identificando 17 elementos principales que se incluyen en una constancia. Se utilizó *MySQL Server*, como sistema gestor de bases de datos y *MySQL Workbench* para el diseño de la base de datos, así como el *framework* de *Laravel* en conjunto con el lenguaje PHP para el desarrollo de modelos, *middleware* y controladores. Además de incorporar PHP con *VueJS* para la creación de las vistas, lo que permitió aportar lógica a las mismas mediante el uso de *JavaScript*, logrando el desarrollo de 3 módulos principales, en los que se encuentran la obtención del documento digital en formato PDF y su validación utilizando códigos QR, así como un módulo enfocado a las tareas del coordinador. Con el desarrollo de este proyecto, se logra cumplir el objetivo de automatizar el proceso de emisión de constancias de estudiante inscrito para el IIT de la UACJ, con un alto potencial

para expandirse en beneficio de más de 33,999 estudiantes actualmente inscritos en la UACJ.

Palabras claves: Constancias electrónicas, automatización de procesos, Proceso Unificado Racional (RUP), Lenguaje Unificado de Modelado (UML), códigos QR.

Abstract

Any technological development focused on adequate information management, brings great benefits to organizations and institutions that bet on them, this research aimed at automating the process of issuing proof of enrollment letter for students in the Institute of Engineering and Technology (IIT) of the *Universidad Autónoma de Ciudad Juárez* (UACJ). For which, direct interviews were conducted with 10 coordinators of educational programs, as well as 9 secretaries of the different departments of the IIT, in addition, a total of 50 student records were analyzed. For the development of the system prototype, the RUP (Rational Unified Process) methodology was used, where 8 use cases, 6 BPMN diagrams for the modeling of the processes, 3 domain models and navigation diagram were used, together, establish the requirements of the electronic constancy system prototype, identifying 17 main elements that are included in a proof of enrollment letter. MySQL Server was used as a database management system and MySQL Workbench for the database design, as well as the Laravel framework in conjunction with the PHP language for the development of models, middleware and controllers. In addition to incorporating PHP with VueJS for the creation of the views, which allowed to provide logic to them through the use of JavaScript, achieving the development of 3 main modules, which include obtaining the digital document in PDF format and its validation using QR codes, as well as a module focused on the coordinator's tasks. With the development of this project, the objective of automating the process of issuing proof of enrollment letter for students for the IIT of the UACJ is achieved, with a high potential to expand for the benefit of more than 35,000 students currently enrolled in the UACJ.

Keywords: Electronic proof of enrollment letter, process automation, Rational Unified Process (RUP), Unified Modeling Language (UML), QR codes.

Introducción

El uso de la tecnología para la automatización de procesos, como una herramienta para la adecuada gestión de información, trae grandes beneficios a las empresas o instituciones que apuestan por este tipo de desarrollos, esto puede lograrse mediante la implementación de sistemas de información y bases de datos, entre otras soluciones específicas. Esta investigación tiene como objetivo automatizar uno de los procesos dentro del Instituto de Ingeniería y Tecnología de la UACJ, enfocado a la emisión de constancias de estudiante inscrito, con lo que se espera contribuir en sentar un precedente tanto dentro de la misma institución, como en otras instituciones en la región, además de lograr que este servicio esté disponible en cualquier momento para los estudiantes. En el capítulo 1, se abordan los antecedentes que sirvieron como base para plantear el problema, además de definir el objetivo y justificación. Las metodologías, herramientas, instrumentos y definiciones que dan fundamento a este proyecto de investigación, se abordan en el capítulo 2. En el capítulo 3, se describe el desarrollo del proyecto, así como la metodología utilizada para la realización del mismo. Los resultados obtenidos después del desarrollo, se muestran en el capítulo 4, en donde, además, se interpretan los resultados desde un punto de vista en cumplimiento con el objetivo planteado. En el capítulo 5, se concluyen los resultados obtenidos respecto al objetivo de esta investigación, además, se brindan recomendaciones a futuro para la continuación de este proyecto o investigaciones relacionadas.

Capítulo 1

Planteamiento del Problema

El crecimiento de la población estudiantil que ha experimentado la Universidad Autónoma de Ciudad Juárez en sus diferentes institutos en los últimos años, ha incrementado consecuentemente los procesos académicos y administrativos. Algunos de estos procesos se realizan de forma manual y no se encuentran estandarizados incluso dentro de un mismo instituto. Uno de ellos es, la emisión de constancias de estudiante inscrito, el cual, no cuenta con un procedimiento establecido para la unificación en las diferentes coordinaciones de programa.

Este proceso surgió como resultado del requerimiento de diferentes dependencias públicas y privadas, las cuales, solicitan a los estudiantes constatar que se encuentran inscritos en una institución de educación superior, presentando un documento emitido por la misma, con el fin de realizar diversos trámites ante estas dependencias.

1.1. Antecedentes

Dentro de las universidades que han realizado proyectos similares de automatización en México, se encuentran el Instituto Politécnico Nacional y la Universidad Autónoma del Estado de Hidalgo, sumándose a universidades extranjeras como la Universidad de Valparaíso, la Universidad de Huelva, la Universidad de Carabobo y la Universidad Francisco Valdivia,

en la búsqueda de brindar servicios de calidad a sus estudiantes.

No solo universidades han optado por mejorar sus servicios automatizando la emisión de documentos manuales a electrónicos, el gobierno federal coordinándose con los gobiernos estatales, ha automatizado el trámite de emisión de actas de nacimiento a través de la información asentada en el Registro Civil, poniendo al alcance de la población la realización de dicho trámite, los 365 días del año y las 24 horas del día, beneficiando a setecientas mil personas en los primeros catorce días del lanzamiento el 10 de enero del 2018 [1, 2].

Existen múltiples propuestas de desarrollo tecnológico similares, presentadas ante instituciones educativas nacionales y extranjeras, tal es el caso de la tesis “Análisis y diseño del sistema de control de servicio social de la UAEH”, presentada en la Universidad Autónoma del Estado de Hidalgo, por las alumnas Claudia Ávila y Marina Ramírez, para obtener el título de Licenciado en Sistemas Computacionales, en donde, se tiene como objetivo automatizar los procesos manuales para dar celeridad a los trámites requeridos en el servicio social, mediante el uso de un sistema de información electrónico [3].

Otra de las propuestas de automatización de servicios, fue la presentada ante el Instituto Politécnico Nacional (IPN), en el trabajo de tesis “Aplicación para emisión de constancias electrónicas”, desarrollada por Efrén Zarazúa Escalona, para recibir el grado de maestría en ciencias en informática, la cual, se centra en la automatización de un proceso interno de la Secretaría de Investigación y Posgrado del IPN, que consiste en la expedición de constancias de productos derivados de la investigación [4].

En el periodo de agosto a diciembre del 2018, se tuvo la oportunidad de realizar una estancia de movilidad estudiantil en la Universidad de Valparaíso, Chile, en sus sistemas de información, cuentan con un módulo de “Portal de Usuario” accesible para los estudiantes con sus credenciales institucionales, desde este módulo, pueden generar y descargar una constancia en formato PDF, este documento cuenta con un folio que permite verificar la autenticidad y fecha de validez ante cualquier dependencia que así lo requiera, a través de

un módulo de validación accesible desde Internet.

1.2. Definición del problema

La Universidad Autónoma de Ciudad Juárez, a través de las diferentes coordinaciones de los programas educativos, ofrece para sus estudiantes, la emisión de una carta constancia de estudiante inscrito en la institución, para diferentes trámites en que esta es requerida.

Para este servicio, el estudiante debe acudir personalmente a la coordinación de su programa educativo, registrar su solicitud y esperar durante un periodo que va de uno a tres días hábiles, antes de acudir nuevamente a recoger la constancia solicitada.

Durante el periodo mencionado, la secretaria del jefe de departamento, revisa los datos proporcionados por el estudiante cotejándolos contra el Sistema de Información Integral (SII v2) en el módulo de Control Escolar o documentos probatorios presentados, si los datos son correctos, elabora la constancia solicitada y la pasa al coordinador de carrera correspondiente para que la firme, regresando el documento a la secretaria, quien coloca el sello del departamento para que finalmente sea entregado al estudiante. Todo este proceso se realiza de forma manual.

1.2.1. Problemas identificados

A continuación, se describen algunos de los problemas identificados en el proceso manual.

- Subutilización de recursos

Se genera una carga de trabajo extra para la secretaria, quien debe realizar el cotejo de los datos proporcionados por el estudiante, contra sistema o documentos probatorios, elaborar la constancia manualmente en el caso de que los datos sean correctos y entregarla al coordinador de carrera.

Se genera una carga de trabajo extra para el coordinador de carrera, quien debe revisar y firmar constancias solicitadas por estudiantes del programa.

En promedio, se realizan diariamente alrededor de 35 trámites de constancias de estudiante inscrito en los cuatro departamentos del IIT, esto se traduce a más de cinco mil constancias emitidas anualmente en hoja membretada, lo que representa un gasto significativo para la institución.

- Tiempo invertido.

Es necesario que los estudiantes, acudan al menos dos ocasiones a su coordinación para finalizar el trámite de constancia de estudios, una para solicitarla y otra para recoger el documento generado. Lo que afecta de forma negativa la satisfacción del alumnado hacia la realización del trámite.

- Falta de seguimiento

Dado que el proceso se realiza de forma manual, si alguno de los actores del proceso falla por causas diversas (no presentarse, seguimiento inadecuado, extravío de formas, etc.), el trámite no se realizará o se demorará más de lo necesario.

- Error humano

Durante el 2018, en las coordinaciones del departamento de física y matemáticas, se presentaron alrededor de 15 casos, en donde las constancias emitidas tuvieron errores ortográficos o de datos, derivados de la captura manual de la información.

- Sólo en horarios y días hábiles

Uno de los problemas cuando se requiere un trámite urgente, es que este, solo pueda ser solicitado en horarios y días hábiles.

1.3. Objetivos

En esta sección, se describen tanto el objetivo general como los objetivos específicos del proyecto de investigación.

1.3.1. Objetivo general

Desarrollar un prototipo de sistema para automatizar la emisión de constancias de estudiante inscrito en el Instituto de Ingeniería y Tecnología de la UACJ.

1.3.2. Objetivos específicos

- Analizar el proceso actual de emisión de constancias.
- Determinar los requerimientos del sistema.
- Modelar las bases de datos del sistema.
- Diseñar, desarrollar e implementar el prototipo del sistema.
- Evaluar el rendimiento del sistema.

1.4. Justificación

La información es considerada como un activo de suma importancia para cualquier institución, por lo que, cualquier desarrollo tecnológico que se enfoque a un manejo adecuado de la información, traerá grandes beneficios sociales, económicos y administrativos.

Con el desarrollo de esta investigación, se beneficiará a más de cinco mil estudiantes inscritos en el IIT, pudiéndose expandir a más de treinta y dos mil estudiantes inscritos en

los diferentes institutos de la UACJ, en donde las estadísticas en el último año (diciembre de 2017 a diciembre de 2018) indican un crecimiento del 4.55 %, pudiendo alcanzar un total de estudiantes beneficiados a más de treinta y cinco mil en los próximos 5 años, al mismo tiempo el proyecto puede sentar un precedente para otras instituciones educativas en la región [5, 6].

En cuanto a la subutilización de recursos, se podrá reducir drásticamente el empleo de hojas membretadas para la emisión de constancias de estudiante inscrito, que al año suman más de cinco mil tan solo en el IIT, lo que representaría un ahorro considerable en cuanto al uso de papel y tóner para la institución. Acorde con esto, también se beneficiará al personal administrativo, quienes dejarán de elaborar las constancias de forma manual.

Algunos casos en los que una constancia sea requerida por el estudiante de manera urgente, ya sea para poder recibir atención médica en un hospital público, tramitar un permiso de viaje ante el consulado de los EE.UU., o recibir un descuento en el pasaje de autobús durante periodos vacacionales, entre otros casos, el trámite estará disponible 24/7 los 365 días del año de manera virtual, por lo que el estudiante no tendrá que preocuparse por horarios, días festivos o periodos vacaciones, para poder generar una constancia de estudiante inscrito.

Las constancias electrónicas, contarán con un certificado digital y un código de seguridad impreso, este último, podrá ser validado a través de Internet, por cualquier dependencia para la que sea dirigida la constancia, desde un módulo especial del sistema, cuya dirección URL estará visible en el documento.

Implementar un sistema de información que permita la gestión de constancias electrónicas, sin duda alguna, será una plataforma que impulse el desarrollo de otros sistemas de información que automaticen distintos procesos dentro de la UACJ, optimizando y estandarizando a la universidad, para lograr una mayor eficiencia y un uso de recursos adecuado.

Capítulo 2

Marco teórico

En esta sección se incluye la teoría y los diversos conceptos y definiciones que dan fundamentación al proyecto de investigación y la revisión minuciosa de fuentes de información.

2.1. Metodologías de análisis, modelado de procesos y aplicaciones

En esta sección, se definen algunos conceptos, metodologías y herramientas, para modelar de procesos y aplicaciones.

2.1.1. Definición de metodología

Una metodología, es una guía que permite llevar a cabo las acciones necesarias de una investigación. Indica qué realizar y el cómo actuar cuando se realiza una investigación. Se define como el enfoque para observar total y sistemáticamente el problema [7].

Análisis: Es el proceso de clasificación e interpretación de hechos, determinación de problemas y uso de información con el fin de recomendar e implementar mejoras [7].

Proceso: Una serie definida de tareas o actividades distribuidas en un espacio de tiempo,

con un inicio, un fin y con materias primas y productos claramente definidos. El modelado de procesos, es una técnica para comprender y describir el proceso [8].

El modelo de un sistema informático se crea en un lenguaje de modelado. Este modelo cuenta con notación y semántica, además, puede convertirse en distintos formatos que incluyan gráficos y texto. La intención es que pueda ser utilizado de una forma más simple para propósitos específicos [9].

2.1.2. Herramientas de modelado de procesos

En esta parte, se describen las herramientas para el modelado de procesos y los diagramas utilizados para el mismo fin.

UML

Unified Modeling Language (UML por sus siglas en inglés) es uno de los primeros métodos de diseño y análisis orientados a objetos, desarrollado a finales de los años ochenta y a principios de los noventa [10].

UML es un lenguaje que permite la construcción, especificación, visualización y documentación de los componentes de un sistema informático o de software. Utiliza herramientas para modelar visualmente y facilitar la administración de modelos, permitiendo mostrar u ocultar detalles cuando sea preciso. Modelar visualmente, favorece la conservación de estabilidad entre los componentes del sistema. Además, mejora la capacidad del equipo de trabajo para administrar la complejidad del sistema [3].

Diagramas de casos de uso

Un caso de uso, ejemplifica la forma de interactuar entre un usuario y un sistema informático, captando las funcionalidades. Se genera al hablar con los diferentes usuarios, analizando en conjunto las diferentes acciones que pretendan realizar con el sistema [10].

UML, adicionalmente emplea los casos de uso como elementos esenciales para el desarrollo de software, utiliza un esquema para representarlos gráficamente, llamado diagrama de casos de uso. En este, se indican las funcionalidades que deberá contar el sistema, pero no detalla cómo servirlos [10, 11].

El objetivo de un diagrama de casos de uso, es modelar las distintas funcionalidades de un sistema desde la percepción de los usuarios, con la intención de incluir todos los casos de uso y actores, demostrando su interacción entre sí [9].

La Figura 2.1 ejemplifica la interacción de usuarios o actores para una taquilla. Los casos de uso se muestran en círculos dentro de un recuadro que representa el sistema [9].

Actores de proceso

Se emplea la palabra actor para referirse a un usuario del sistema, quien participa en los casos de uso. En el diagrama, los actores se representan con la figura humana, pero también pueden ser sistemas externos que interactúan con los casos de uso [10].

Modelos BPM

Business Process Modeling (BPM por sus siglas en inglés) es una rama de la ingeniería basada en modelos, centrada en modelar los distintos procesos de negocio en una organización. Un BPM describe los requerimientos del sistema a construir [12].

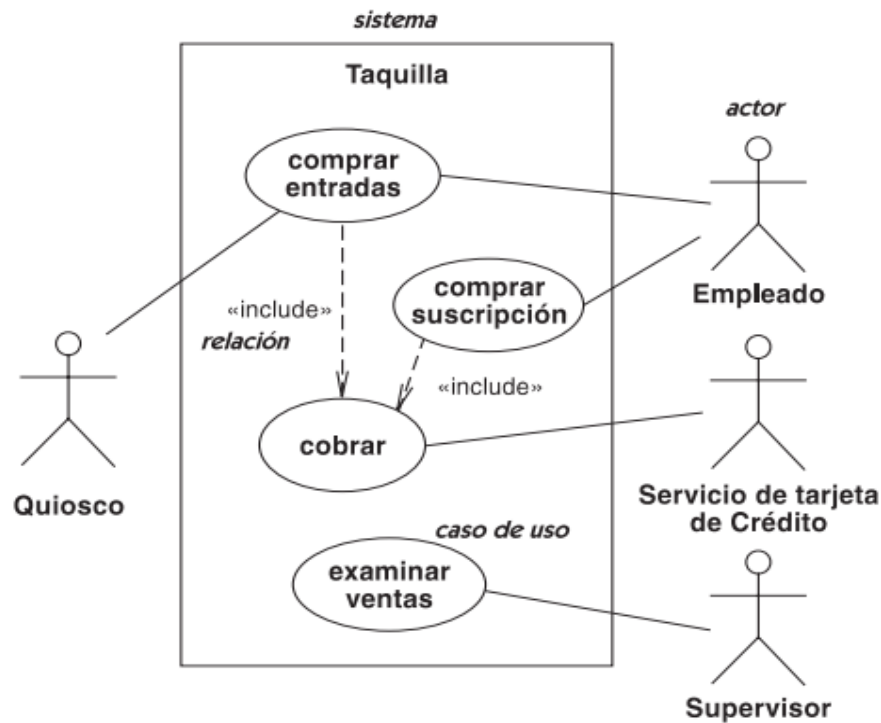


Figura 2.1: Diagrama de casos de uso [9].

Diagramas BPMN

Business Process Modeling Notation (BPMN por sus siglas en inglés) se refiere a la descripción gráfica que representa la lógica en un proceso en la organización. Diseñada para gestionar la sucesión de mensajes y procesos entre los participantes. Provee un lenguaje sencillo para una eficiente comunicación de los procesos entre las partes involucradas [13].

En la Figura 2.2, se ejemplifica un proceso con diversos elementos que representan el cómo un proceso se comporta, se incluyen actividades, compuertas de decisión y eventos de inicio y finalización, todos ellos conectados por líneas secuenciales que representan el flujo [13].

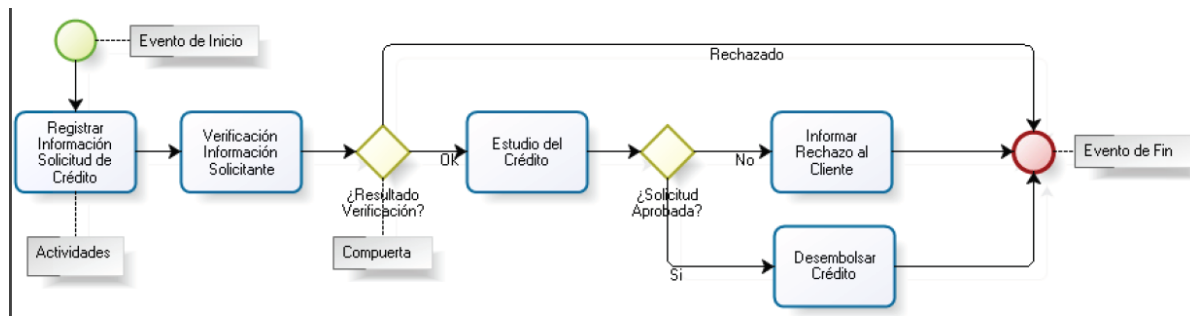


Figura 2.2: Diagrama BPMN [13].

Diagramas de estados

Este diagrama, modela los diferentes estados de un objeto, cada estado representa un lapso en la vida del objeto, cuando se satisfacen condiciones específicas. Si un evento se suscita, puede producir una transición que haga cambiar dicho estado. Estos diagramas se emplean para representar interfaces, sistemas reactivos o pasivos que adoptan diferentes fases durante su tiempo de vida, cada una con comportamientos singulares [9].

La Figura 2.3 ejemplifica con recuadros los distintos estados para un objeto, indicando con un punto el estado inicial y las diferentes transiciones de acuerdo a las acciones descritas sobre las flechas [9].

2.1.3. Herramientas de modelado de aplicaciones

En este apartado, se definen los diagramas para el modelado de aplicaciones.

Diagramas de flujo de datos

Los diagramas de flujo de datos (DFD) tienen la intención de registrar el flujo y los procesos a los que se somete la información en una organización. Estos representan las funciones

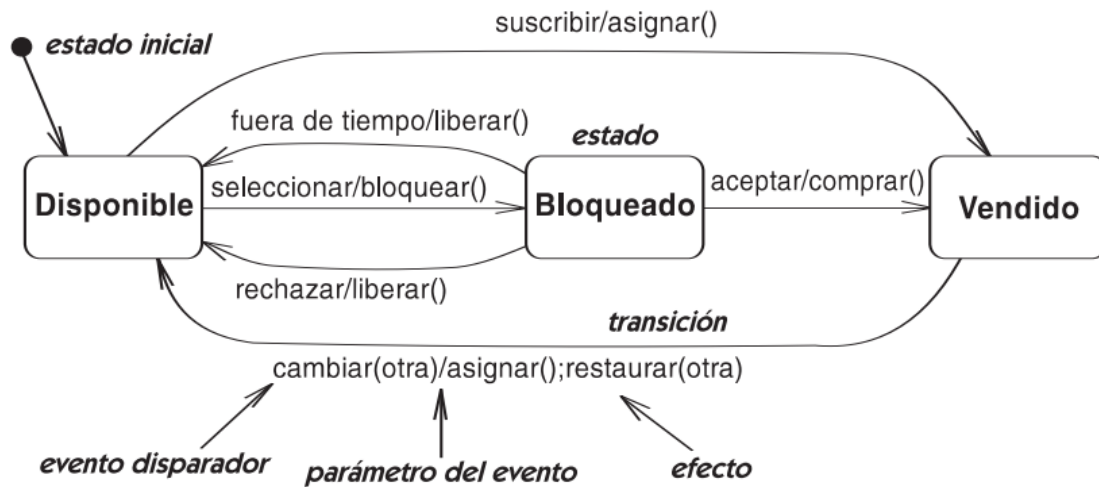


Figura 2.3: Diagrama de estados [9].

del procesamiento de datos, datos instaurados o utilizados por el sistema de procesamiento o personas. Los DFD pueden ser representados en capas. En la Figura 2.4, se muestran los símbolos elementales que representan los DFD [3, 14].



Figura 2.4: Símbolos básicos de los DFD [14].

Diagrama entidad-relación

Este diagrama fue propuesto para el diseño de bases de datos de tipo relacional, en el que se identifican componentes elementales como lo son, atributos, relaciones e indicadores. Su objetivo primordial es la representación de objetos o entidades y sus distintas relaciones. Una

entidad, representa los datos o atributos que componen la información que será empleada por el sistema. Una relación, es la forma en que se asocian las distintas entidades [3, 15].

Diagrama conceptual de clases

Este diagrama se utiliza para el diseño de conceptos de los casos de uso, con el propósito de saber cómo se ajustan en el sistema. Los rectángulos representan las clases, estos también contienen los atributos y métodos asociados a dichas clases. Las líneas que enlazan los rectángulos representan los diferentes tipos de relaciones [9, 10].

La Figura 2.5, ejemplifica un diagrama de clases para un modelo de “venta de entradas”, mostrando las clases importantes y las distintas relaciones entre las mismas [9].

Modelo de dominio

Este tipo de modelo es una representación gráfica de las clases u objetos en un dominio en particular. Es frecuentemente utilizado para el diseño de los objetos del sistema, siendo uno de los modelos más importantes creados durante el análisis [16].

La Figura 2.6, ejemplifica un modelo de dominio, en donde se incluyen las clases más significativas para el dominio específico, así como sus atributos y relaciones [16].

Diagrama de secuencia

El diagrama de secuencia se utiliza para mostrar los roles de los objetos o individuos que interactúan con el sistema en un caso específico, se utiliza una línea doble vertical, para representar el tiempo de vida del objeto. El diagrama de secuencia describe mensajes, pero sin definir el tiempo en que suceden. El mensaje es representado por una flecha que conecta los diferentes objetos. La Figura 2.7 ejemplifica un diagrama de secuencia para un sistema de compra de entradas [9].

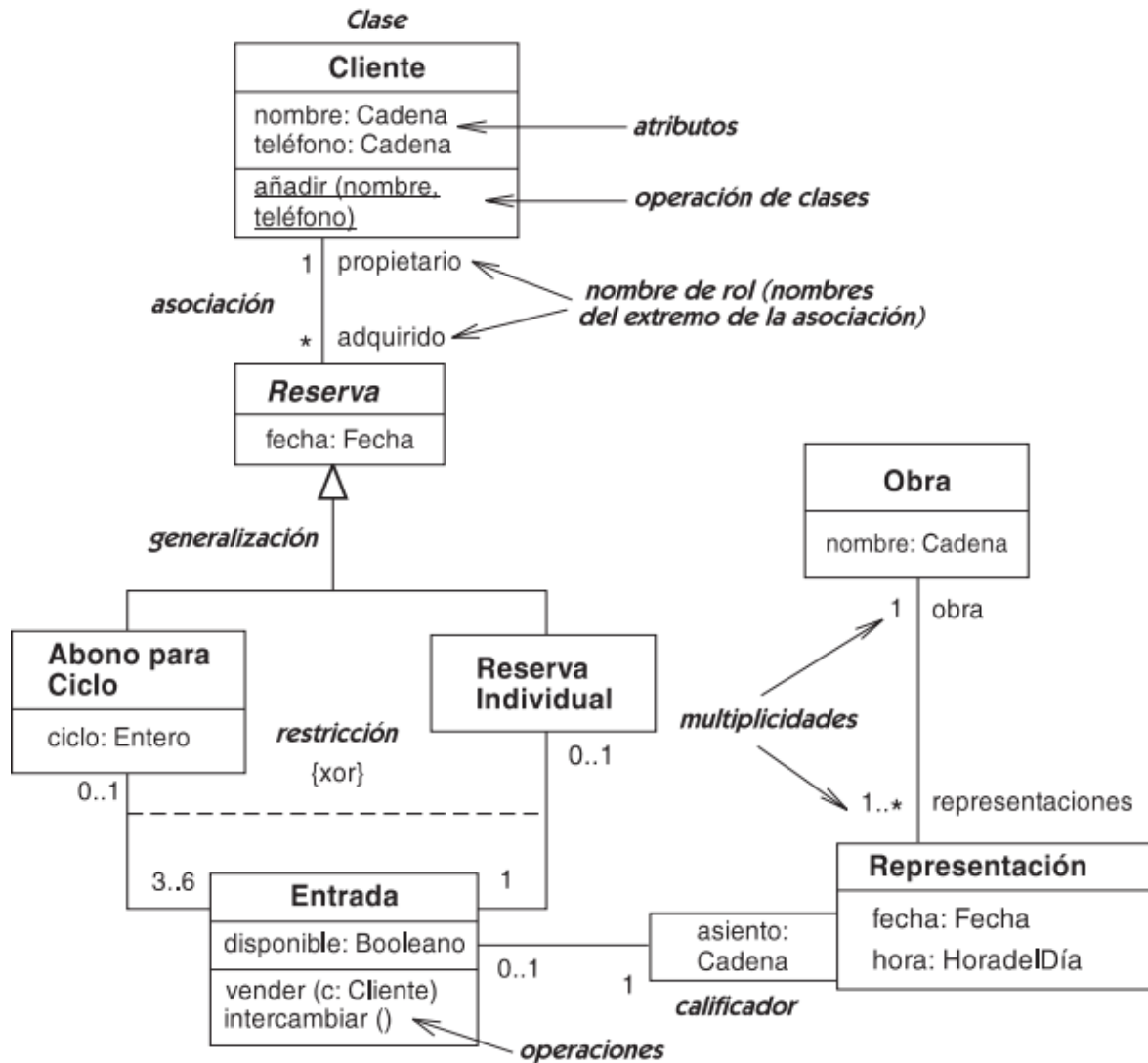


Figura 2.5: Diagrama de clases [9].

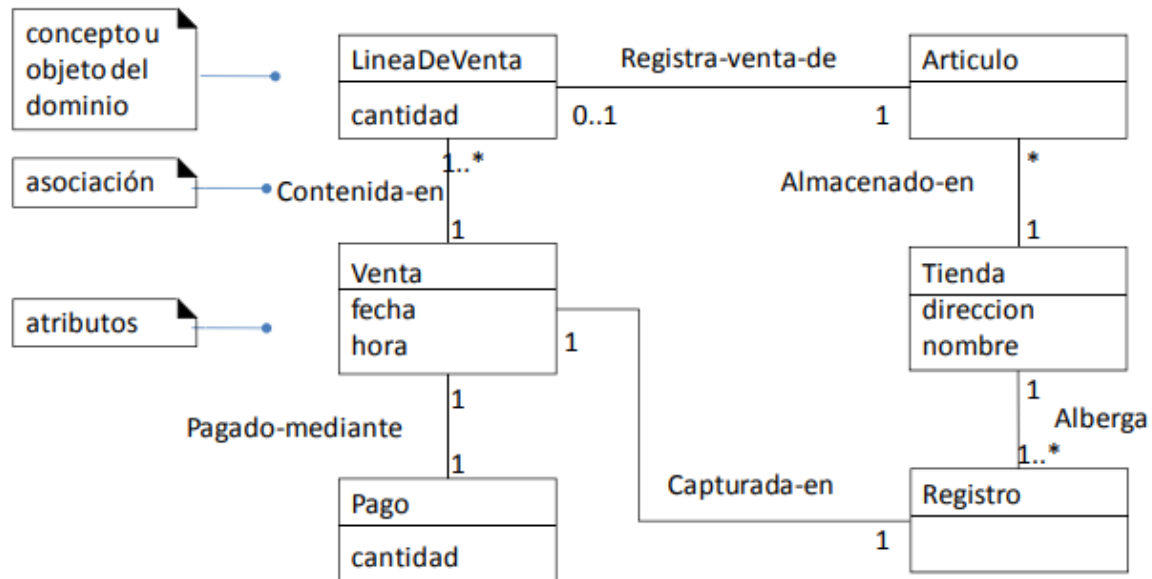


Figura 2.6: Modelo de dominio [16].

Prototipo de sistema

Un prototipo representa de forma simple un sistema. Una vez creado el prototipo, puede ser descartado o completado hasta lograr el desarrollo completo del sistema. Se centra en entregar muestras rápidas de cómo se verá un sistema [17].

2.2. Bases de datos

En esta sección, se define el concepto de bases de datos, clasificaciones, modelos y los sistemas de gestión.

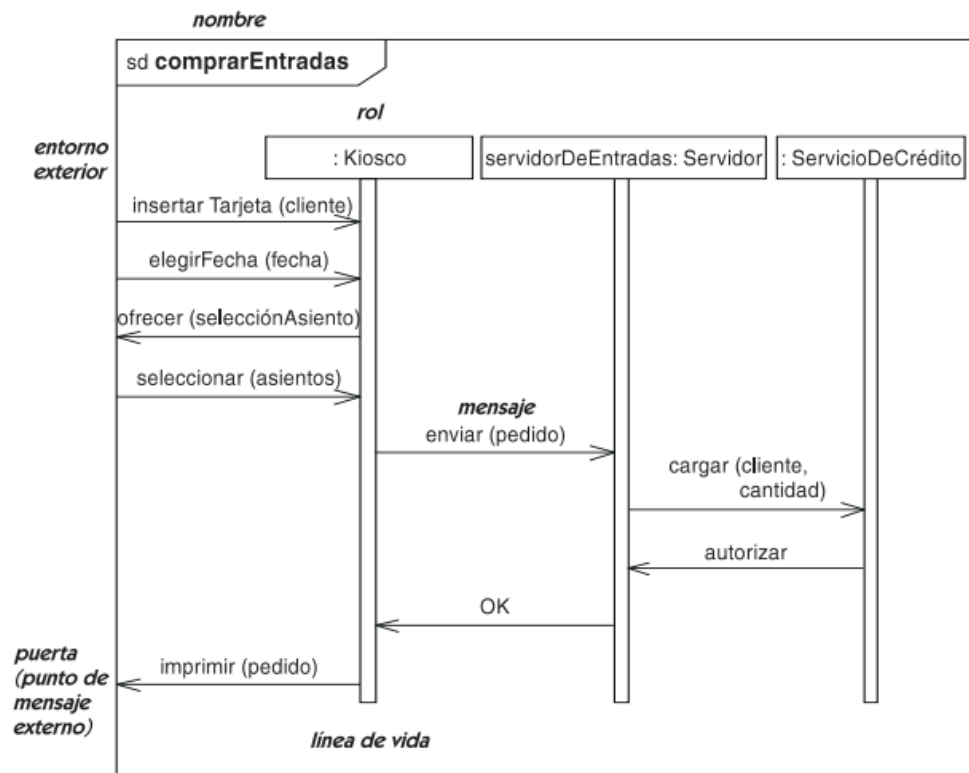


Figura 2.7: Diagrama de secuencia [9].

2.2.1. Definición de base de datos

Una colección de registros relacionados da lugar a una base de datos, que representa de forma integral las entidades y sus relaciones. Esta representación de datos estructurados, debe permitir utilizarse por diferentes usuarios de forma conjunta [18, 19].

2.2.2. Clasificación de bases de datos

Las distintas bases de datos, pueden ser clasificadas en diferentes maneras, de acuerdo al contexto, utilidad o a las necesidades que deban satisfacer [20].

Bases de datos estáticas

Son únicamente de lectura, principalmente empleadas para el registro de datos históricos que puedan ser utilizados a futuro para, analizar el comportamiento de los datos al pasar del tiempo, efectuar proyecciones, toma de decisiones o realizar un estudio de la información [20].

Bases de datos dinámicas

En estas, la información que se almacena es modificada con el paso del tiempo, permiten rutinas de actualización, edición y eliminación de datos, además de consulta [20].

Bases de datos bibliográficas

Contienen información básica de la fuente principal que permita localizarla, almacenando información del autor, editorial, título, etc., al igual que un resumen de la publicación, no así la totalidad del texto [20].

Bases de datos de texto completo

Utilizadas para almacenar la totalidad del texto de las fuentes primarias [20].

2.2.3. Modelos de bases de datos

Un modelo de bases de datos es una abstracción que representa el contenedor de datos que almacena la información, además de los métodos asociados para guardar o recuperar dicha información. Las bases de datos pueden ser modeladas de acuerdo a su función y al modelo de gestión de datos [20].

Bases de datos jerárquicas

Este modelo organiza los datos en forma de árbol, donde los nodos padres pueden tener varios hijos o raíces, los nodos que no tienen hijos son conocidos como hojas. Este modelo permite la creación de estructuras con un excelente rendimiento, especialmente cuando las aplicaciones manejan un volumen de información considerable [20].

Base de datos de red

Su diferencia en cuanto al modelo jerárquico, es el permitir que un nodo cuente con más de un padre, lo que representa una mejora para los problemas en los que exista redundancia de datos, aunque la administración de la base de datos tiene mayor dificultad [20].

Bases de datos transaccionales

Diseñadas para el envío y recepción de datos que requieran una gran rapidez de transferencia, comúnmente empleadas para aplicaciones de análisis de datos. Debido a que su

enfoque es la velocidad, la redundancia o duplicidad de información no representan un problema como en otros modelos de bases de datos, suelen estar conectadas a bases de datos relacionales para aprovechar al máximo su rendimiento [20].

Bases de datos relacionales

Este modelo de bases de datos, contiene tablas estructuradas en filas y columnas, las cuales están relacionadas por un campo en común llamado identificador. El grado está determinado por la cantidad de columnas y la cardinalidad es el número de registros [4, 21].

Bases de datos no relacionales

Son aquellas que no utilizan el esquema tabular de filas y columnas. Utilizan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena. Están optimizadas particularmente para sistemas que requieran grandes volúmenes de información o modelos de datos con flexibilidad [22, 23].

2.2.4. Sistemas de gestión de bases de datos (SGDB)

Son sistemas que administran el acceso y gestión de los datos, como un intermediario entre los archivos y los usuarios, para hacerse cargo de la explotación, comprobación y mantenimiento de los datos. Permitiendo al usuario tratar la información mediante el uso de un lenguaje conceptual [21].

MySQL

Es un sistema gestor de bases de datos relacionales, almacena una gran cantidad de datos de tipos diversos, enfocado principalmente a cubrir las necesidades de organizaciones,

2.3. LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DISEÑO DE APLICACIONES WEB

organismos administrativos y empresas [18].

Microsoft SQL Server

Es el sistema gestor de la empresa Microsoft, para bases de datos relacionales. Contiene características y herramientas variadas, que se pueden emplear para el desarrollo y administración de base de datos y soluciones que se basen en ellas. Basado en lenguaje Transact-SQL, brinda rendimiento y protección de datos para clientes de aplicaciones embebidas y aplicaciones web, entre otros. Capaz de administrar grandes cantidades de datos simultáneamente [4].

2.3. Lenguajes de programación y herramientas de diseño de aplicaciones web

En este apartado, se definen algunos de los lenguajes de programación y herramientas, para el diseño y desarrollo de aplicaciones web.

2.3.1. Definición de lenguaje de programación

Un lenguaje de programación es un lenguaje comprensible a los humanos, para escribir programas de computadora. Las computadoras solo entienden datos binarios, por lo que un compilador debe traducir el programa desde un lenguaje de programación a código binario ejecutable [4].

2.3. LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DISEÑO DE APLICACIONES WEB

2.3.2. PHP

Acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje interpretado de alto nivel, principalmente empleado para el desarrollo web al poder ser incrustado en páginas HTML. Tiene como propósito permitir la creación de páginas dinámicas de forma fácil y rápida, su código se ejecuta del lado del servidor [24].

2.3.3. NodeJS

Es una plataforma desarrollada sobre el entorno javascript para la construcción de aplicaciones de red, de forma rápida y escalable. Utiliza un modelo de entrada y salida dirigido por eventos, lo que lo hace eficiente y ligero, ideal para aplicaciones en tiempo real con un intensivo uso de datos. Proporciona un conjunto de librerías para la creación de aplicaciones enfocadas a redes de comunicaciones, como parte de estas librerías, se incluyen funciones para comunicarse con el sistema operativo [25].

2.3.4. Javascript

Es un lenguaje enfocado a ejecutarse del lado del cliente, en la capa de presentación de aplicaciones web, ligado a la interfaz de programación para la gestión de estilos de componentes, contenido y estructura de páginas web, con el fin de proporcionar al usuario una experiencia interactiva. Forma parte del paradigma de programación orientada a eventos, donde el flujo del programa es asíncrono y dependiente de eventos [25].

2.3.5. HTML

HyperText Markup Language (HTML por sus siglas en inglés), utiliza etiquetas para estructurar el contenido de las páginas web. No es considerado un lenguaje de programación

2.3. LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DISEÑO DE APLICACIONES WEB

sino un lenguaje de etiquetado, donde generalmente las etiquetas son utilizadas en pares, para apertura y cierre, declarando el contenido entre ellas [26].

2.3.6. CSS

Cascading Style Sheets (CSS por sus siglas en inglés), trabajan en conjunto con HTML para proporcionar estilos visuales a los componentes del documento, cambiando la forma en que estos se presentan, como color, tamaño, bordes [26].

2.3.7. MVC

Es un patrón arquitectónico para el desarrollo de aplicaciones, basado en el modelo, vista y controlador, que aísla la lógica de negocio de la interfaz de usuario, lo que facilita la actualización, escalabilidad y mantenimiento de los componentes, aumentando la flexibilidad y reutilización de los mismos. El modelo gestiona los datos y sus transformaciones, la vista administra la representación visual al usuario e interactúa con el controlador, este último, es el objeto que maneja las órdenes del usuario y opera sobre los datos presentados por el modelo [27, 28].

2.3.8. *Bootstrap*

Bootstrap es una herramienta que permite crear interfaces de usuario adaptables a pantallas de cualquier tamaño, ofreciendo herramientas para adaptar la visualización de los diferentes componentes de las vistas en páginas o sitios web, utilizando paquetes de librerías CSS, las cuales incluyen todo tipo de elementos empleados en el diseño y presentación [29].

2.4. Metodologías de desarrollo de software

En esta sección, se definen los conceptos de metodologías de desarrollo de software, así, como los enfoques generales y específicos de algunas de ellas.

2.4.1. Definición de metodología

Una metodología es un marco de referencia de trabajo, con el fin de ser utilizada para guiar las tareas y actividades en el desarrollo de software. Esta forma de trabajo otorga las herramientas necesarias para establecer un proceso de desarrollo de software ordenado y eficiente, indicando los pasos a seguir, las reglas y las restricciones que se deberán cumplir durante el proceso [30].

2.4.2. Ciclo de vida del software

Durante el ciclo de vida del software, existen actividades primordiales que se han de llevar a cabo durante su desarrollo. Estas actividades incluyen el especificar los requisitos que deberá cumplir el sistema, elaborar el análisis de dichos requisitos, su diseño e implementación, incluir pruebas e integración de los diferentes componentes, mantenimiento y documentación [31].

En la siguiente lista, se definen las distintas actividades que deberán realizarse a lo largo del ciclo de vida del software [31]:

- **Especificación de requisitos.** En esta etapa se definen los requerimientos del sistema a desarrollar. Donde el equipo de desarrollo negocia con el cliente, además que favorece la planeación y control del proceso de desarrollo.
- **Análisis.** Esta actividad se centra en entender y comprender el porqué de los requisitos, con el fin de elaborar una arquitectura adecuada del sistema.

- **Diseño.** Una vez obtenida la arquitectura del sistema, esta se transforma en una arquitectura especializada, en ella, se considera el entorno donde será implementado el sistema.
- **Implementación.** Es donde la arquitectura especializada, que fue desarrollada en la etapa de diseño, se traduce a un lenguaje que pueda ser entendido y procesado por la computadora, es el código programado.
- **Integración.** Se integran los componentes desarrollados de forma independiente, para completar la construcción del sistema.
- **Pruebas.** Es la etapa de verificación y validación, tanto de los componentes individuales, como del sistema completo. Suele ser uno de los puntos más importantes en el desarrollo y debe realizarse en conjunto con las demás actividades, en busca de errores o defectos. Las pruebas son desarrolladas en distintos niveles, que van desde funciones simples hasta el sistema entero.
- **Documentación.** Se documentan los aspectos más importantes de todas las etapas anteriores, con el fin de ayudar a los usuarios y encargados del mantenimiento y actualización del sistema desarrollado.
- **Mantenimiento.** Esta etapa se centra en corregir errores que no hayan sido detectados durante el desarrollo o pruebas del sistema.

2.4.3. Metodologías de desarrollo tradicionales

El objetivo principal de estas metodologías tradicionales, es separar las funcionalidades de un sistema, con el fin de expresar en términos de funciones, la totalidad del sistema, especificando los datos de entrada y salida que correspondan a cada funcionalidad [31].

RUP

Rational Unified Process (RUP por sus siglas en inglés), es un proceso de desarrollo de software que brinda disciplina en la asignación de responsabilidades y tareas en un equipo de desarrollo. La meta de esta metodología de desarrollo, es ayudar a la elaboración de sistemas de alta calidad, que se adapten a los requerimientos de los usuarios finales, con costos y calendario de actividades previsibles. RUP, pretende integrar todas las facetas a considerar durante el ciclo de vida del software, con el propósito de poder ser utilizado por proyectos de diversos tamaños [32].

Sus características principales, residen en ser una metodología que se centra en la arquitectura, es iterativa, incremental y conducida por casos de uso. Una iteración en RUP, puede ser realizada a través de una cascada, atravesando las etapas primordiales del ciclo de vida del software. Al término de cada iteración, se elabora la integración de resultados con otros obtenidos de iteraciones preliminares. En la Figura 2.8, se muestra el proceso de iteración en RUP [3].

El proceso es dividido en cuatro etapas, dentro de estas, se ejecutan diversas iteraciones que varían en cantidad y esfuerzo según sea el proyecto o la etapa en la que se encuentre. La Figura 2.9, ejemplifica la variación del esfuerzo ejercido en cada una de las disciplinas [3].

2.4.4. Metodologías de desarrollo ágiles

Son sistemas de administración de proyectos enfocados a utilizar el tiempo de forma efectiva. Suelen ser útiles para la visualización y organización de tareas e incrementar el trabajo en equipo y su rendimiento. Permiten llevar un seguimiento a detalle de cada una de las etapas del proyecto, ya sea personal o grupalmente [33].

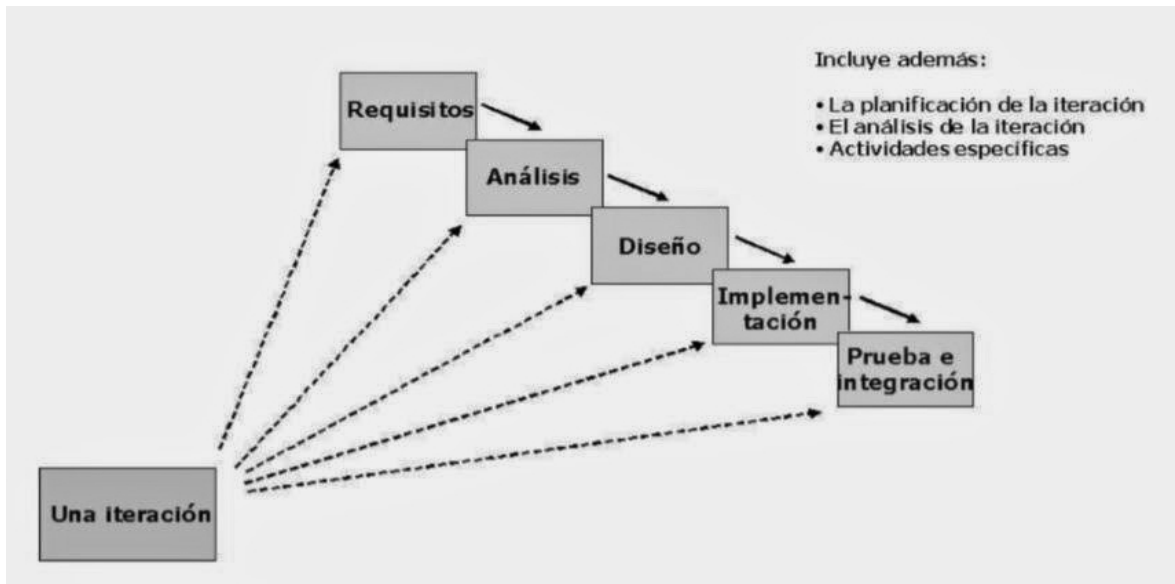


Figura 2.8: Iteración RUP [3].

Scrum

Esta metodología surge con el propósito de gestionar de forma dinámica los proyectos de desarrollo de software. Plantea trabajar en iteraciones o *sprints* sobre entregas parciales, al iniciar cada iteración, se definen las tareas se llevarán a cabo y al finalizar se entregan resultados precisos. Propone dividir el equipo de trabajo en roles específicos para fomentar la colaboración. En la Figura 2.10, se resume el proceso de desarrollo de Scrum [33, 34].

2.5. Técnicas e instrumentos de recolección de datos

Estas técnicas e instrumentos de recolección de datos se refieren al establecimiento de los medios precisos utilizados para su recolección, el tipo de instrumento, el grado de medición, así como su validez y confiabilidad [35].

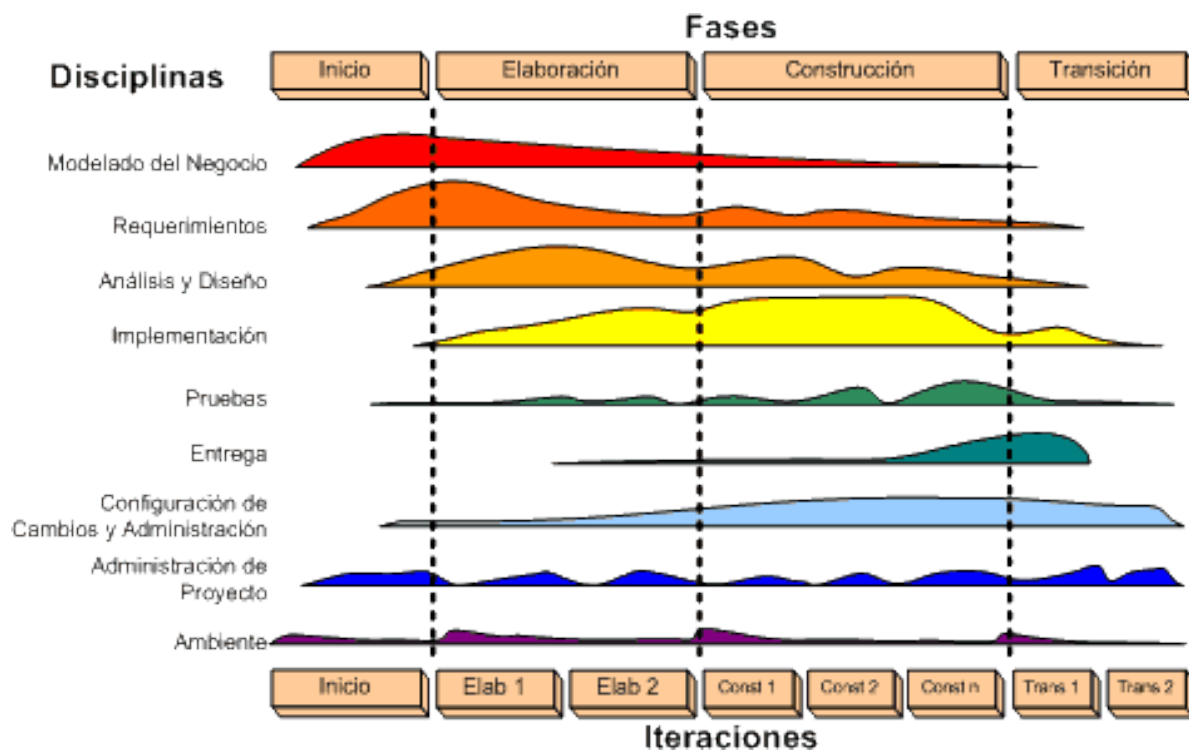


Figura 2.9: Variación de esfuerzo de acuerdo a la fase del proyecto [3].

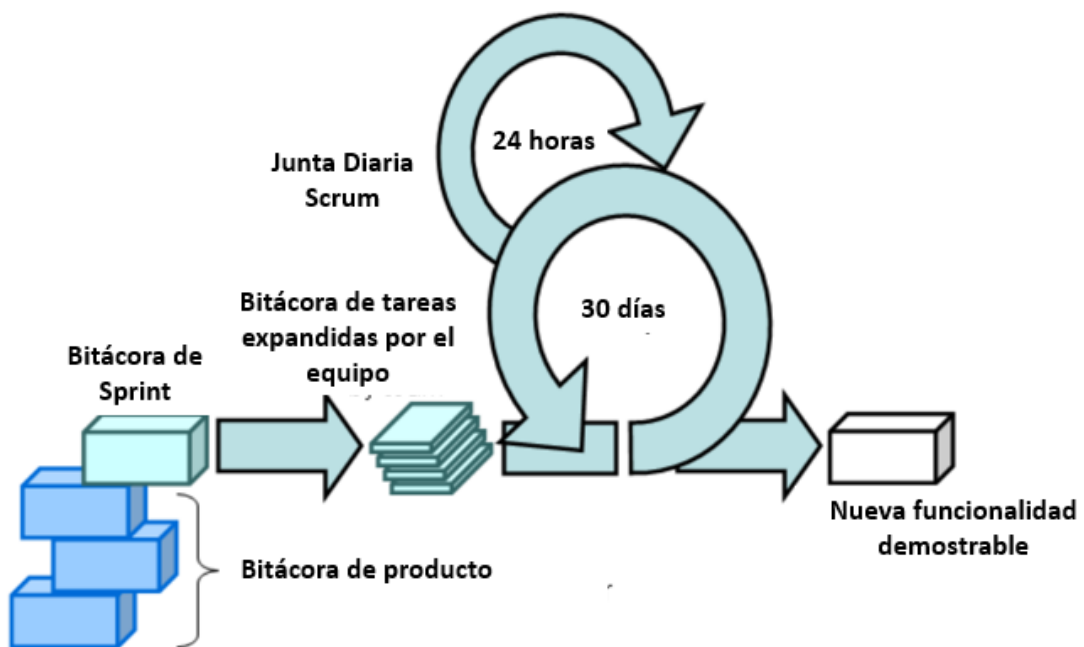


Figura 2.10: Esquema de proceso Scrum [34].

2.5.1. Cuestionario

Las respuestas que se obtienen a través de cuestionarios utilizando preguntas cerradas son cuantificables. Las respuestas a preguntas abiertas pueden ser evaluadas o ponderadas empleando el análisis e interpretación. Los cuestionarios permiten formular las preguntas en el contexto de satisfacción o insatisfacción con el sistema. Además, se podrían sugerir algunas características del sistema para preguntar a los interlocutores y que recuerden cuáles características son de interés [35].

2.5.2. Validación de instrumento

La validación se refiere al nivel en que el resultado obtenido de un instrumento, refleja verdaderamente lo que está siendo medido por este [35].

2.5.3. Confiabilidad de instrumento

Se define como la falta de errores aleatorios en algún instrumento de recolección de datos, como resultado, si los puntajes o resultados que se obtienen son semejantes, existe confiabilidad. En instrumentos con respuestas dicotómicas, puede aplicarse el coeficiente KR20, en donde el instrumento es dividido en partes. La Figura 2.11, describe la fórmula para calcular el índice de confiabilidad del instrumento y la clasificación de la escala obtenida [35].

2.6. Seguridad de la información

En esta sección, se definen los conceptos de seguridad de la información, así como los servicios, la planificación, tecnologías y estándares.

$$k_{r20} = \frac{k}{k-1} \left[1 - \frac{\sum p^* q}{(v)^2} \right]$$

$k_{r20} =$	Coefficiente de confiabilidad.	ESCALA	CATEGORÍA
$K =$	Número de ítems que contiene el instrumento	0 – 0,20	Muy baja
$\sum p^* q =$	Sumatoria de la varianza individual de los ítems.	0,21 – 0,40	Baja
$V =$	Varianza total de la prueba.	0,41 – 0,60	Moderada
		0,61 – 0,80	Alta
		0,81 – 1	Muy Alta

Figura 2.11: Coeficiente KR20 [35].

2.6.1. Concepción de la seguridad de la información

La seguridad de la información se basa en medidas reactivas y preventivas que empresas u organizaciones emplean para el resguardo y protección de los datos, con el fin de conservar la confidencialidad, disponibilidad e integridad de estos. Es necesario tener en cuenta la seguridad de la información, durante todo el ciclo de vida del software [15, 36].

Confidencialidad

La confidencialidad evita que la información sea divulgada a sistemas o personas no autorizadas [36].

Disponibilidad

Es la propiedad de la información, de encontrarse disponible a entidades para acceder a ella, ya sean aplicaciones, sistemas, procesos o personas, siempre y cuando estas, cuenten con

la autorización necesaria en el momento de la disposición [37].

Integridad

La integridad tiene como objetivo conservar la información libre de modificaciones no autorizadas, haciendo que su contenido se mantenga sin alteraciones, a menos que se modifique por personas autorizadas [37].

2.6.2. Planificación de la seguridad

La planeación de seguridad tiene como propósito, brindar una visión amplia de los requerimientos de seguridad, describiendo los controles que servirán para dar cumplimiento a esos requisitos. Esta planificación, también debe incluir las responsabilidades y las conductas que se esperan de todas las personas con acceso al sistema [37].

2.6.3. Tecnologías

En este apartado, se definen algunas de las tecnologías relativas a la seguridad de la información.

Cortafuegos

Es un muro que protege las redes internas y seguras de las redes que no se consideran de confianza. El cortafuegos provee un punto controlado para la comunicación entre redes, comúnmente denominado punto de estrangulamiento [38].

SSL (capa de sockets segura)

Suministra conexiones seguras a través del cifrado de los datos que se intercambian entre un servidor y un cliente, autenticando al servidor. Utilizar SSL puede reducir el rendimiento de la red, ya que impacta en la velocidad al tener que cifrar la información que se transmite [38].

Certificado digital

Tiene como objetivo validar la identidad de quien posee el certificado, de manera muy parecida a como lo hace un pasaporte. Un tercero de confianza, llamada autoridad certificadora (CA), es quien se encarga de emitir dichos certificados digitales para servidores y usuarios [38].

2.6.4. Estándares de seguridad de la información

En esta sección, se señalan algunos de los estándares referentes a la seguridad de la información.

ISO/IEC 27000

Esta norma brinda una visión más amplia del conjunto de normas que forman parte de la serie 27000, indica el alcance, actuación y la intención de la publicación de cada una de ellas. Contribuye a formar las bases de la importancia de implementar un sistema de gestión de seguridad de la información (SGSI) [39].

ISO/IEC 27001

Cuenta con los requerimientos del sistema de gestión de seguridad de la información. Sintetiza los objetivos de control y los controles que son desarrollados en la norma ISO 27002:2005, con el fin de que las organizaciones puedan seleccionarlos para el desarrollo de sus SGSI [39].

ISO/IEC 27002

Tiene la intención de ser una guía de buenas prácticas, describiendo los controles recomendados para la seguridad de la información. Cuenta con 133 controles y 39 objetivos de control, los cuales se agrupan en 11 dominios [39].

2.7. Usabilidad

El término nace a partir del desarrollo de Internet. El objetivo de este paradigma, es lograr que las aplicaciones cuenten no solo con las funcionalidades para las cuales fueron diseñadas, sino también, puedan atraer al usuario y facilitarle el trabajo mediante interfaces de fácil uso [40].

2.7.1. Conceptos sobre usabilidad

En el siguiente listado, se definen algunos conceptos básicos de la usabilidad [40]:

- **Usabilidad.** Es la cualidad del software que permite a un usuario determinado, conseguir los objetivos propuestos en un contexto de uso específico, con satisfacción y efectividad.

- **Comprensibilidad.** Característica que otorga al usuario la capacidad de comprender si el software es apropiado y la forma en la que puede ser empleado, para realizar tareas en situaciones de uso determinadas.
- **Aprendizaje.** Es la característica del software, que permite al usuario aprender a utilizarlo, en un contexto de uso específico.
- **Operabilidad.** Característica que permite al usuario conseguir operar y controlar el software, en situaciones específicas de uso.
- **Atractividad.** Capacidad del software para ser atrayente a los usuarios. En cuanto a propiedades del software, tales como diseño, gráficos y color.
- **Conformidad a estándares y pautas.** Se refiere a la medida en que el software se adhiere a los estándares, guías de estilo o regulaciones afines a la usabilidad.

2.7.2. Estándares de usabilidad

Se define como estándares de usabilidad a los requerimientos, recomendaciones o pautas, basados en principios demostrados en la práctica. Constituyen un convenio de un conjunto de profesionales legalmente autorizados a nivel internacional, nacional o local [41]. A continuación, se mencionan algunos estándares referentes a la usabilidad.

ISO/IEC 9126-1 FDIS

Destaca a la usabilidad como una propiedad de la calidad del software, que asocia la evaluación del diseño, la interacción y la interfaz del usuario. Considera la usabilidad en términos de un software comprensible, operable, atractivo y de conformidad con los estándares de usabilidad [40].

ISO 9241-11:2018

Es el estándar concerniente a los requerimientos ergonómicos para trabajar con terminales de presentación visual. Las labores de oficina que incluyen software ofimático y procesamiento de datos, son cubiertas por este estándar [41].

ISO 13407

Este estándar, provee una guía para las tareas de diseño que se centran en las personas a lo largo del ciclo de vida del software interactivo. Incorporando técnicas, elementos humanos y ciencias ergonómicas, con el propósito de obtener un software eficiente y efectivo, además de mejorar los ambientes de trabajo para las personas [41].

2.7.3. Guías de estilo

Las guías de estilo proporcionan a los desarrolladores de software, un conjunto de normas y principios para basar sus diseños. Esto permite garantizar que las diferentes partes del sistema sean consistentes. Por esta razón es importante que las empresas de desarrollo de software cuenten con una guía a la que puedan adherirse sus desarrolladores [41].

2.7.4. Atributos de la usabilidad

A continuación, se listan los principales atributos de usabilidad en un sistema interactivo [41]:

- **Facilidad de aprendizaje.** Se refiere a la reducción del tiempo necesario para lograr un uso productivo del software, desde el desconocimiento total del mismo.

- **Tiempo de respuesta.** Es la capacidad del software de responder a las solicitudes del usuario. Al depender en parte de las características del equipo de cómputo, este suele ser un factor variable.
- **Flexibilidad.** Esta capacidad del software se refiere a las diferentes maneras de intercambiar información con el usuario. Un sistema con flexibilidad debe brindar control al usuario, ser adaptable y configurable a las necesidades del entorno en que se emplea.
- **Robustez.** Determina la capacidad de otorgar lo necesario para que el usuario pueda cumplir con sus objetivos, además de brindar el asesoramiento necesario.
- **Recuperabilidad.** Determina el grado de facilidad del software para proveer al usuario la forma de corregir una instrucción, cuando esta sea reconocida como errónea.
- **Sintetizabilidad.** Característica del software que permite al usuario percibir cambios en la operación del sistema.
- **Consistencia.** Característica del software de poder ser empleado de la misma forma, sin importar el momento en que sea utilizado.
- **Disminución de la carga cognitiva.** Reducir la carga cognitiva en interfaces interactivas, se refiere a proveer al usuario aspectos que este, pueda reconocer sin necesidad de memorizarlos. Este concepto determinará el diseño de los diferentes elementos interactivos que se mostrarán en la interfaz.

2.7.5. Evaluación de la usabilidad

Evaluar la usabilidad es pieza clave en el desarrollo de sistemas que posean alguna interfaz o interacción con el usuario. Cuando no se realiza alguna evaluación, no es posible conocer si el sistema satisface los requerimientos de los usuarios o si se ajusta apropiadamente en el

contexto de la organización en el que será empleado. Existen métodos propuestos para evaluar la usabilidad, que tienen como objetivo realizar una medición de los distintos elementos que se relacionan con esta. Seleccionar un método depende de diferentes factores, tales como, recursos necesarios, espacio para el desarrollo de pruebas y tecnologías empleadas [41].

Capítulo 3

Desarrollo del Proyecto

En este capítulo, se aborda el producto propuesto y su funcionamiento, se describe la metodología utilizada para el desarrollo del mismo, además, se expone el análisis realizado sobre los procesos actuales de emisión de constancias de estudiante inscrito para los 4 diferentes departamentos del IIT, describiendo cada uno de forma detallada mediante diagramas BPMN, siendo esta una de las partes fundamentales, que permitieron determinar los requerimientos del sistema, el modelado de las bases de datos y el diseño del prototipo. También, se describen las etapas del desarrollo del producto propuesto, hasta llegar al entregable final, siendo este, la constancia electrónica.

3.1. Producto propuesto

Se propuso el desarrollo de un prototipo de sistema automatizado de emisión de constancias electrónicas de estudiante inscrito, el cual, está dividido en tres módulos principales, como se muestra en la Figura 3.1.

- **Módulo de coordinadores.** Este módulo está dirigido para ser utilizado por los coordinadores de programas educativos, quienes ingresan utilizando su correo electrónico institucional a través de la pantalla de inicio de sesión, tiene como funcionalidades,

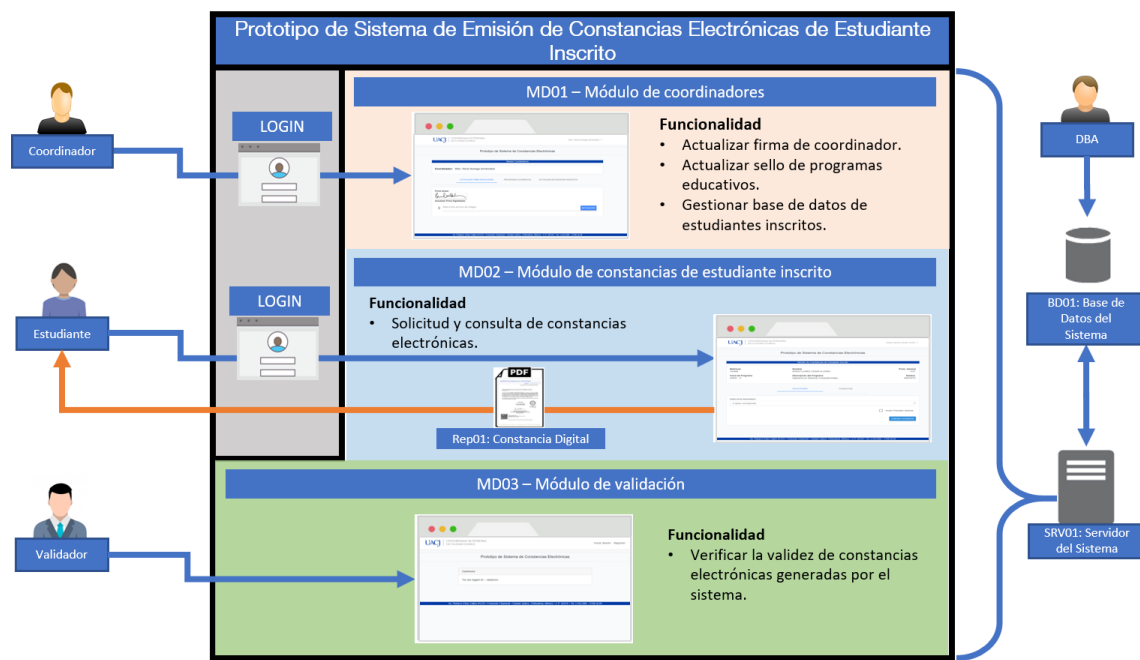


Figura 3.1: Diagrama conceptual del prototipo de sistema de emisión de constancias de estudiante inscrito. (Elaboración propia).

actualizar la firma personal en formato digital de cada coordinador y el sello digitalizado del programa educativo, mismos que serán colocados en la constancia digital. Además, este módulo provee al coordinador la funcionalidad de gestionar los registros de estudiantes que se encuentren inscritos en su programa.

- **Módulo de constancias de estudiante inscrito.** Este módulo, está dirigido a los estudiantes inscritos, quienes tendrán acceso utilizando su correo electrónico institucional, tiene como funcionalidad el permitir solicitar y generar las constancias electrónicas, también, el estudiante podrá consultar las constancias generadas con anterioridad y descargarlas nuevamente.
- **Módulo de validación.** La funcionalidad del módulo de validación, es permitir que los encargados de las dependencias, a quienes se remiten las constancias digitales generadas en el sistema, puedan constatar mediante un código de validación único, incrustado en cada documento, la autenticidad de los datos y la validez de la constancia digital.

El administrador de la base de datos (DBA), es el encargado de la gestión de usuarios que ingresan al sistema, así como de la actualización de los datos necesarios para su funcionamiento, tales como, los relacionados con programas educativos, fechas de inicio y fin de los periodos semestrales, fechas de periodos vacaciones, coordinadores y dependencias a las que son dirigidas las constancias digitales.

3.2. Descripción de la metodología

Después de haber analizado las bases teóricas de las distintas metodologías, se determinó utilizar la metodología RUP para la realización de este proyecto, por tener un proceso dirigido a casos de uso y a la arquitectura del sistema, además, por ser orientada a objetos y utilizar UML como parte integrante.

3.2.1. Análisis del proceso actual de emisión de constancias

Para analizar el proceso actual de emisión de constancias en las diferentes coordinaciones de los programas académicos del IIT, como primer paso, se entrevistó a secretarias de los jefes de departamento y a coordinadores, quienes son los encargados directos de proveer el servicio de emisión de constancias de estudiante inscrito, para los cuatro departamentos.

Las preguntas realizadas en las entrevistas, fueron no sólo con la finalidad de comprender el proceso de emisión de constancias de estudiante inscrito, sino también, de recabar datos estadísticos, tales como, la cantidad de constancias emitidas al año, los periodos en los que se presenta una mayor demanda, las distintas dependencias a las que estas van dirigidas, además, de determinar los datos que se incluyen en cada constancia. Derivado de la información obtenida, se desarrollaron cuatro diagramas de modelado de proceso de negocio (BPMN) con la intención de analizar el proceso, mismos que se detallan en los puntos siguientes.

- **Modelado de proceso de negocio para el departamento de física y matemáticas.** En este diagrama (ver Figura 3.2), el proceso inicia cuando el estudiante se presenta en su coordinación para solicitar la constancia de forma verbal, la secretaria del jefe de departamento, le pide al alumno anotar sus datos en un formato (ver apéndice A), después de ello, la secretaria los verifica en el módulo de control académico del sistema institucional SiiV2, los datos que serán validados son: matrícula, nombre completo del estudiante, programa académico y estatus inscrito. Si los datos son correctos, la secretaria elabora la constancia de forma manual desde una plantilla de *Word*, se imprime en hoja membretada y se pasa a firma con el coordinador. El coordinador, además de firmar la constancia, revisa de forma visual los datos impresos y la regresa a la secretaria. Una vez que la constancia cuenta con la firma del coordinador, la secretaria estampa el sello entintado en la constancia y la resguarda para ser entregada al estudiante.

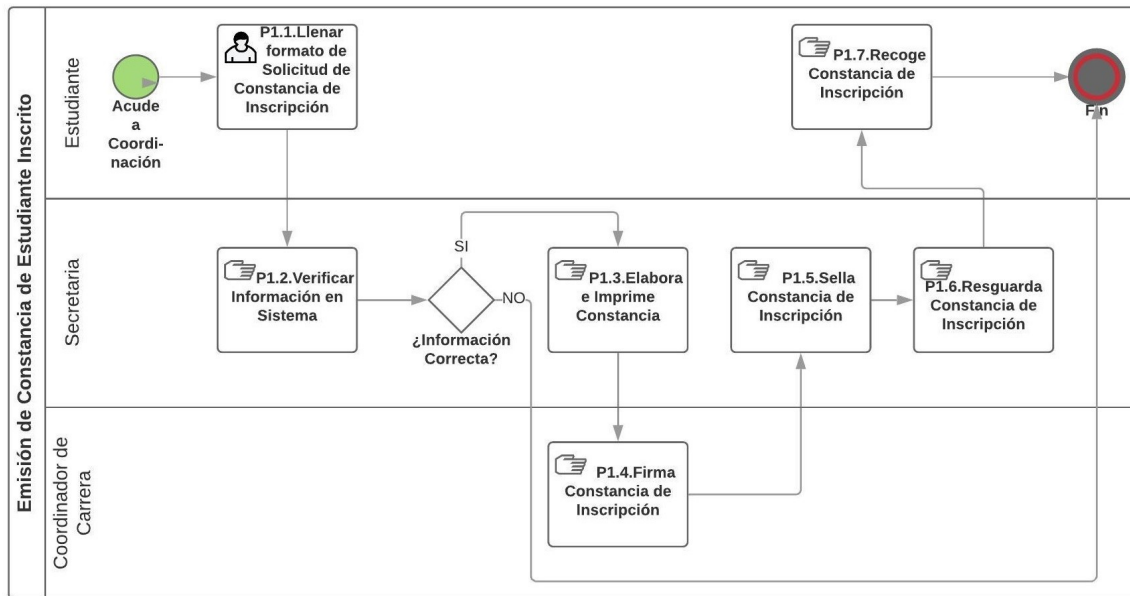


Figura 3.2: BPMN proceso actual departamento física y matemáticas. (Elaboración propia).

- **Modelado de proceso de negocio para el departamento de ingeniería eléctrica y computación.** En este diagrama (ver Figura 3.3), el proceso inicia cuando el estudiante acude a la coordinación para solicitar la constancia de estudiante inscrito, la secretaria del jefe de departamento, anota los datos del estudiante y los relacionados con la constancia e indica al estudiante cuándo deberá presentarse a recoger el documento solicitado. La secretaria verifica la información en el módulo de control académico del SiiV2, si los datos son correctos, elabora e imprime la constancia de forma manual a partir de una plantilla de *Word*, una vez impresa, se la entrega al coordinador, quien revisa los datos visualmente o en sistema, coloca su firma manuscrita y la devuelve a la secretaria para que coloque el sello y se encargue del resguardo. El proceso finaliza cuando el estudiante acude nuevamente a recoger la constancia o si la información proporcionada no es correcta.
- **Modelado de proceso de negocio para el departamento de ingeniería indus-**

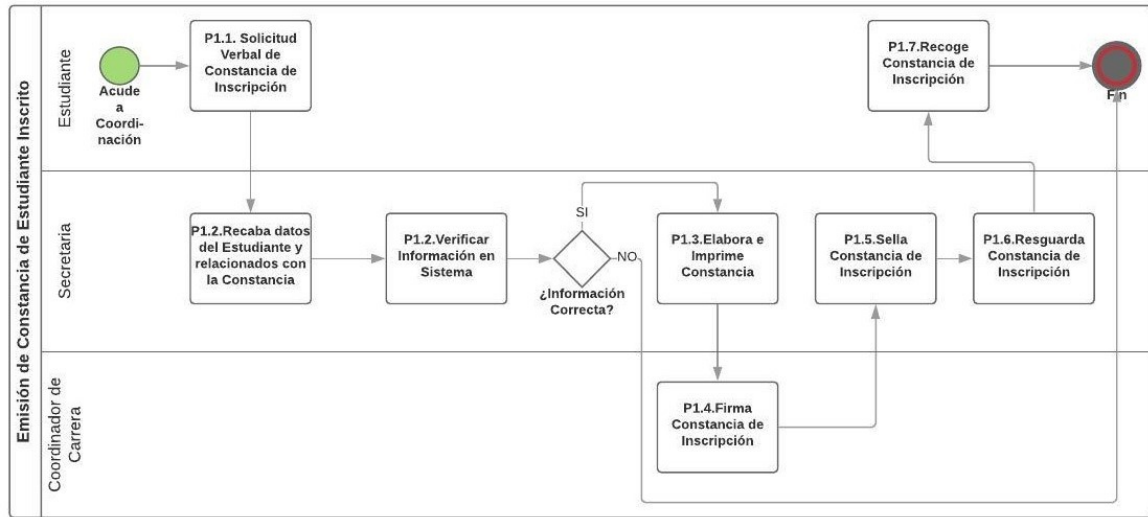


Figura 3.3: BPMN proceso actual departamento ingeniería eléctrica y computación. (Elaboración propia).

trial y manufactura. En este diagrama (ver Figura 3.4), el proceso inicia cuando el estudiante acude a la coordinación a solicitar la constancia, para ello, deberá presentar la tira de materias del periodo en curso y el comprobante de pago de inscripción. La secretaria del jefe de departamento, recaba los datos del estudiante y los relacionados con la constancia a partir de los documentos presentados por el estudiante. Elabora e imprime la constancia de forma manual utilizando una plantilla de *Word*, una vez impresa, se la pasa al coordinador para su firma. El coordinador revisa y firma el documento para después, regresarlo a la secretaria. Esta última, coloca el sello del departamento y se encarga del resguardo. El proceso finaliza cuando el estudiante acude nuevamente a la coordinación a recoger la constancia.

- **Modelado de proceso de negocio para el departamento de ingeniería civil y ambiental.** En este diagrama (ver Figura 3.5), el proceso inicia cuando el estudiante acude a la coordinación a solicitar la constancia, para ello, deberá presentar la tira de materias del periodo en curso. La secretaria del jefe de departamento, recaba los datos

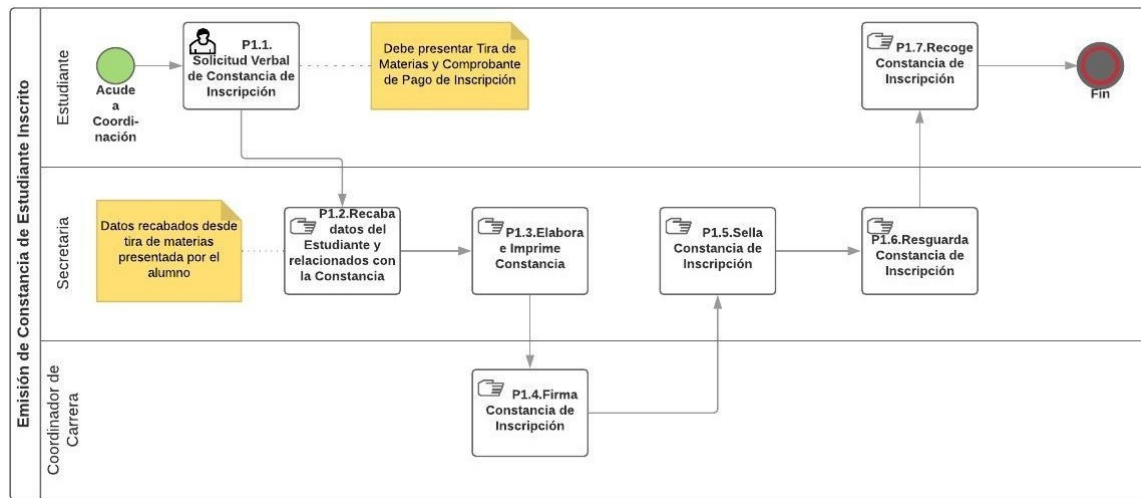


Figura 3.4: BPMN proceso actual departamento ingeniería industrial y manufactura. (Elaboración propia).

del estudiante y los relacionados con la constancia a partir del documento presentado por el estudiante. Elabora e imprime la constancia de forma manual utilizando una plantilla de *Word*, una vez impresa, se la pasa al coordinador para su firma. El coordinador revisa y firma el documento, para después, regresarlo a la secretaria. Esta última, coloca el sello del departamento y se encarga del resguardo. El proceso finaliza cuando el estudiante acude nuevamente a la coordinación a recoger la constancia.

Como resultado de este análisis, se observó que el proceso de emisión de constancias, aún dentro del mismo instituto, en este caso el IIT, no se encuentra actualmente estandarizado entre los cuatro departamentos. También se determinaron los elementos requeridos, utilizados en las constancias de estudiante inscrito.

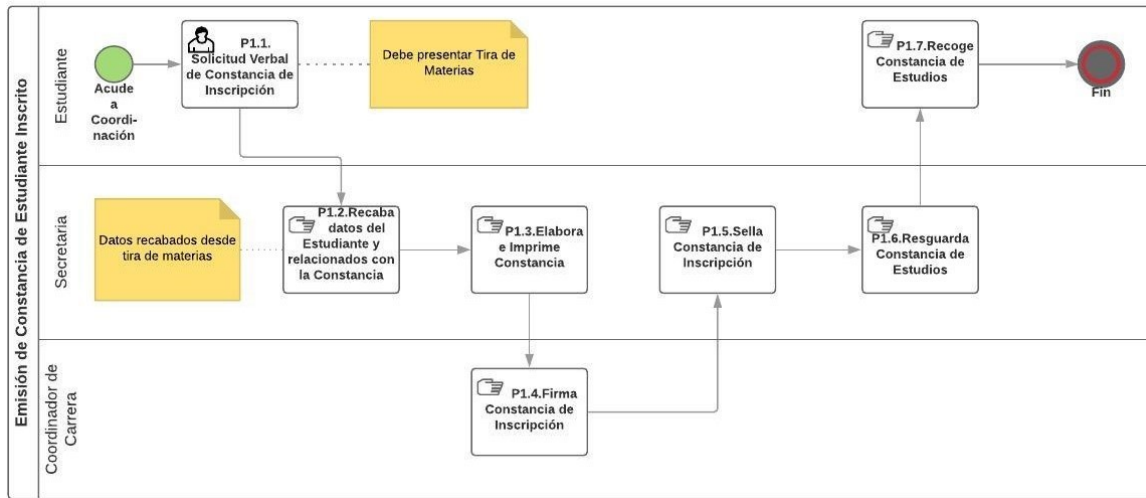


Figura 3.5: BPMN proceso actual departamento ingeniería civil y ambiental. (Elaboración propia).

3.2.2. Determinar los requerimientos del sistema

Para determinar los diversos datos que se incluyen en una constancia de estudiante inscrito, se analizó una muestra de 50 constancias. En la Tabla 3.1, se listan los elementos que se encontraron en los documentos analizados, así como su porcentaje de ocurrencia en la muestra. Con base en los resultados del análisis realizado, se incluyeron los siguientes elementos a las constancias digitales.

- **Membrete institucional y pie de página.** El membrete institucional y el pie de página utilizados en las constancias digitales, generadas en el prototipo de sistema, se extrajeron a partir de la digitalización a 1200dpi de una hoja membretada física en formato JPEG. Estos archivos, no se encuentran almacenados en algún directorio público o privado dentro de la estructura del sistema, sino que fueron codificados en *base64*, para ser incrustados directamente en el documento digital, al momento de generar la constancia en PDF.

Tabla 3.1: Porcentaje de ocurrencia de elementos en constancias analizadas.

Elemento	Ocurrencia	Porcentaje
Membrete Institucional	50	100 %
Fecha de emisión	50	100 %
Destinatario Dependencia	50	100 %
Nombre del estudiante	50	100 %
Matrícula del estudiante	50	100 %
Programa educativo	50	100 %
Semestre	48	96 %
Promedio general	5	10 %
Periodo actual	50	100 %
Fecha de inicio de periodo	43	86 %
Fecha de fin de periodo	43	86 %
Periodos vacacionales	43	86 %
Clave institucional	50	100 %
Nombre del coordinador	50	100 %
Firma del coordinador	50	100 %
Sello del departamento	50	100 %
Pie de página (dirección y teléfono)	50	100 %

- **Fecha de emisión.** La fecha de emisión es generada automáticamente por el sistema, configurada para la zona horaria “América/Chihuahua”. Esta fecha de emisión, se almacena en un campo tipo *datetime* en la base de datos, para posteriormente ser consultada y estampada en la generación de la constancia digital.
- **Destinatario.** De acuerdo al análisis de la muestra, se identificaron 10 destinatarios o dependencias principales a las que son dirigidas las constancias. Este listado se muestra en la Tabla 3.2, el listado de dependencias se almacena en una tabla relacionada con las constancias generadas, esto contribuye a generar una estadística, para conocer cuáles dependencias son las que solicitan mayormente el documento a los estudiantes, lo que permite que se tomen las acciones necesarias para informar a estas entidades, el uso de un sistema de emisión de constancias automatizado.

Tabla 3.2: Dependencias principales, identificadas en el análisis de requerimientos.

Nombre de Dependencia
1. A quien corresponda
2. Instituto Mexicano del Seguro Social
3. Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado
4. Instituto Nacional de Migración
5. Secretaría del Trabajo y Previsión Social
6. Secretaría de Educación Pública
7. Secretaría de la Defensa Nacional
8. Coordinación Nacional de Becas de Educación Superior
9. Secretaría de Relaciones Exteriores
10. U.S. Customs and Border Protection

- **Nombre, matrícula y programa educativo del estudiante.** El nombre completo, matrícula y programa educativo del estudiante, son datos que se muestran en la constancia digital, esta información se extrae directamente de una tabla de la base de datos. La tabla, es actualizada por los coordinadores, con la información extraída del sistema institucional SiiV2, e ingresada a la base de datos del prototipo de sistema desde el módulo de coordinadores.
- **Semestre.** Este elemento, se encontró en una ocurrencia del 96 % de la muestra analizada. Por lo que fue imperativo incluir en la constancia digital. Representando uno de los aspectos que se dejarán a discusión, puesto que en un alto porcentaje de los estudiantes inscritos, existe una discrepancia entre el semestre cronológico que se encuentran cursando y el semestre real en relación a las materias cursadas. Por esta razón, para la obtención de este dato, se optó por crear una tabla en la base de datos, en la cual, se incluyeron 10 columnas que representan los semestres, en ellas, se almacenan la cantidad de créditos acumulados para considerar el avance real del estudiante. Además, se incluye una llave foránea, que relaciona estos datos con el programa educativo.

En la Figura 3.6, se muestra la estructura de la tabla que almacena la cantidad de

créditos por semestre, con la que es calculado este dato para la constancia digital. Con base en la cantidad de créditos cursados por el estudiante, consultada en la información proporcionada por el coordinador, extraída del SiiV2.

programas_clavePrograma	primero	segundo	tercero	cuarto	quinto	sexto	septimo	octavo	noveno	decimo
45200 - 11	48	96	148	194	242	290	332	372	388	NULL

Figura 3.6: Cantidad de créditos por semestre del programa de ingeniería en sistemas computacionales. (Elaboración propia).

- **Promedio general.** Este dato, al encontrarse solo en un 10 % de la muestra analizada, se consideró incluirlo de forma opcional para el estudiante, al momento de solicitar la constancia digital. Se extrae desde la información del SiiV2, proporcionada por el coordinador.
- **Periodo actual y periodos vacacionales.** El periodo actual, se almacena en una tabla dentro de la base de datos, guardando la descripción, fechas de inicio y fin, el estatus y la clave del periodo, esta última, como identificador único. Además, está relacionada con los periodos vacacionales correspondientes, que, a su vez, cuentan con un identificador único y la fecha de inicio y final de vacaciones. Lo anterior, permite que puedan existir más de un periodo vacacional por cada periodo semestral.
- **Nombre del coordinador.** El grado académico y nombre del coordinador que se imprime en la constancia digital, es extraído de la tabla de coordinadores en la base de datos, y se relaciona con la clave del programa que cursa el estudiante.
- **Firmas digitalizadas de coordinadores.** Las firmas digitalizadas, son ingresadas al sistema por los coordinadores de programas, desde el módulo de coordinadores. Estas, se almacenan codificadas en *base64*, en un campo de tipo objeto binario grande (BLOB) dentro de la base de datos. Los formatos soportados son TIF, JPEG, PNG y BMP.

La firma es asociada al coordinador mediante una llave foránea, que corresponde al identificador único del usuario que ha iniciado sesión en el módulo correspondiente.

- **Sello digitalizado de coordinaciones.** Al igual que las firmas digitalizadas de los coordinadores, los sellos de los programas educativos, se almacenan codificados en *base64*, en un campo BLOB dentro de la base de datos, con los mismos formatos soportados. El sello es asociado al programa educativo mediante una llave foránea, que corresponde al identificador único de la clave del programa.

3.2.3. Modelado de las bases de datos del sistema

El modelado de las bases de datos para el prototipo de sistema, se realizó a partir de la identificación de los elementos que se incluyen en la constancia digital. Además, de los diferentes actores que interactúan con los módulos establecidos.

Esquema de base de datos

En la Figura 3.7, se muestra el esquema de base de datos, que incluye las tablas empleadas en el prototipo del sistema de emisión de constancias de estudiante inscrito, así como sus campos, tipos de datos y relaciones.

La tabla de alumnos, almacena la información básica del estudiante, como matrícula, correo y nombre. Tiene una relación uno a muchos con la tabla de constancias, lo que permite que un estudiante pueda tener cero o más constancias generadas, y una constancia en particular, pertenezca solo a un estudiante. La tabla de alumnos, también está directamente relacionada con la tabla de programas educativos, donde la relación uno a muchos, asocia una matrícula de estudiante a un solo programa educativo.

La tabla de programas educativos, almacena la clave del programa como identificador único, así como la descripción del mismo. Tiene relación uno a uno con la tabla de sellos, lo

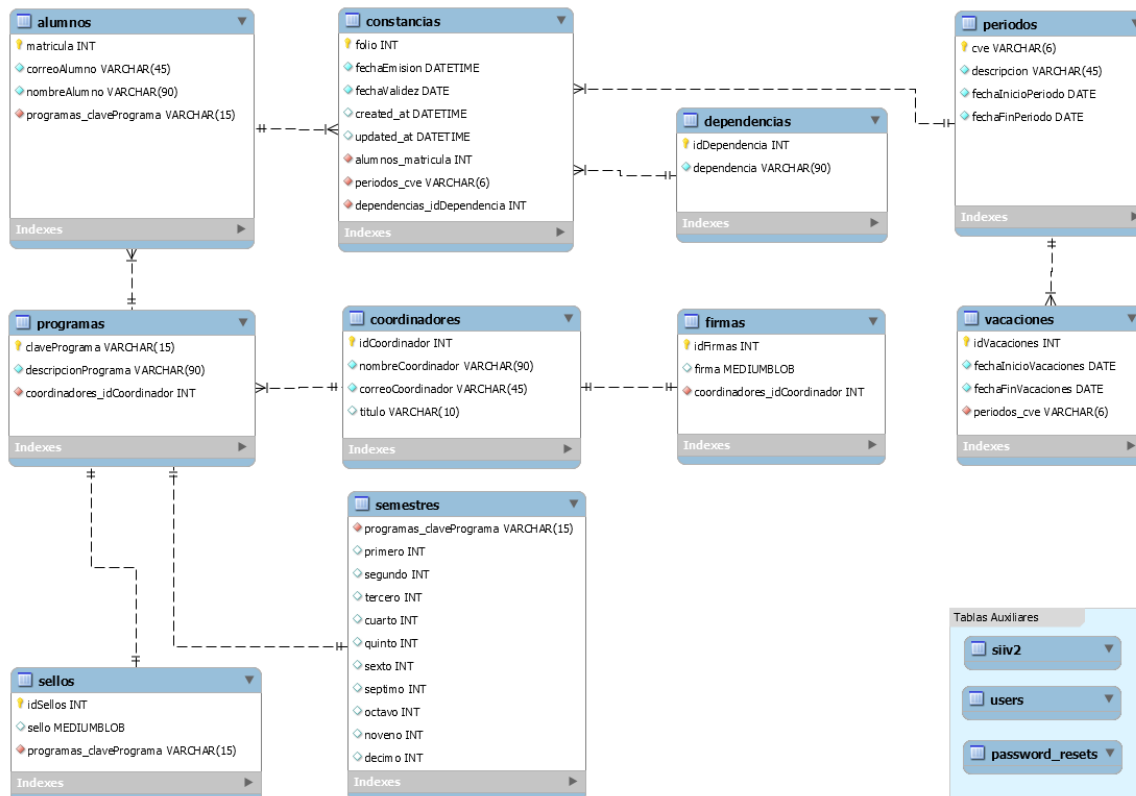


Figura 3.7: Esquema de base de datos del prototipo de sistema de emisión de constancias de estudiante inscrito. (Elaboración propia).

que asocia un sello digitalizado a cada programa educativo. También, cuenta con una relación uno a uno con la tabla de semestres y relación muchos a uno con la tabla de coordinadores, lo que permite que un coordinador pueda tener varios programas a su cargo.

La tabla de coordinadores, almacena el nombre, correo y grado académico del coordinador. Cuenta con una relación uno a uno con la tabla de firmas, lo que asocia una firma digitalizada a cada entidad de coordinador.

La tabla de semestres, fue implementada para almacenar la cantidad de créditos requeridos por semestre en cada programa educativo, es por ello que cuenta con una relación uno a uno, vista anteriormente, con la tabla de programas educativos. Con el uso de esta tabla, fue posible realizar el cálculo del semestre cursado por el estudiante, con base en los créditos y no de forma cronológica.

La tabla de constancias, almacena el folio como identificador único, las fechas de creación, modificación, actualización y validez. Además de la relación con la tabla de alumnos, vista anteriormente, cuenta con relaciones muchos a uno con las tablas de periodos y dependencias. Con ello, cada constancia queda asociada a un solo periodo y a una sola dependencia.

La tabla de dependencias, almacena el nombre de las diez dependencias o destinatarios principales, a los que son dirigidas las constancias. Estas dependencias fueron identificadas en el análisis de requerimientos del sistema.

Las tablas de periodos y vacaciones, almacenan las fechas de inicio y fin correspondientes. Cuentan con una relación uno a muchos, lo que permite que un periodo escolar, pueda tener uno o más periodos vacacionales.

Además de las tablas mencionadas, se utilizan tablas auxiliares que no tienen una relación directa en la base de datos, pero son empleadas desde los diversos controladores de los módulos del prototipo de sistema.

Modelos de dominio

Para el módulo de coordinadores, participan 4 entidades principales, como se muestra en la Figura 3.8. La funcionalidad de este modelo de dominio, es permitir que un coordinador, pueda asociar su firma digitalizada y el sello del programa que coordina.

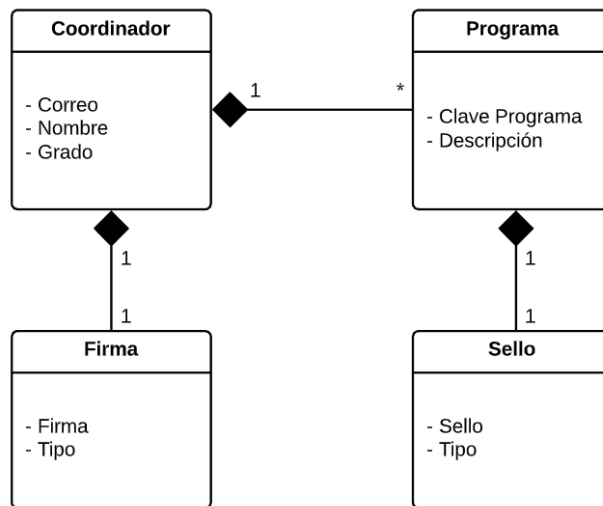


Figura 3.8: Modelo de dominio, módulo de coordinadores. (Elaboración propia).

Para el módulo de constancias de estudiante inscrito, participan 10 entidades principales, como se muestra en la Figura 3.9. La funcionalidad de este módulo, es la generación de constancias digitales de estudiante inscrito, con las relaciones entre las entidades, se logra la obtención de los datos que serán utilizados en la creación de la constancia.

Para el módulo de validación de constancias de estudiante inscrito, participan 4 entidades principales, tal como se muestra en la Figura 3.10. La funcionalidad de este módulo, es constatar la validez de una constancia digital emitida por el sistema, mediante un código de verificación generado por el mismo. La entidad validadora, proporcionará este código, y las entidades en el modelo de dominio, generan los datos que serán mostrados en caso de que la verificación sea correcta.

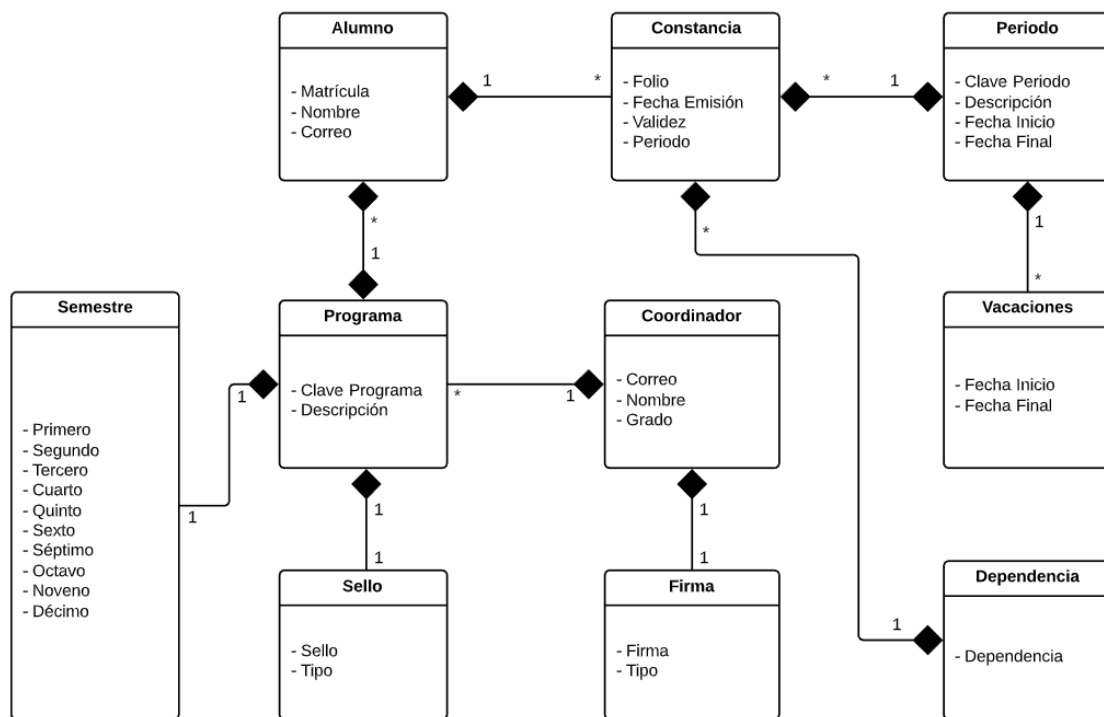


Figura 3.9: Modelo de dominio, módulo de constancias de estudiante inscrito. (Elaboración propia).

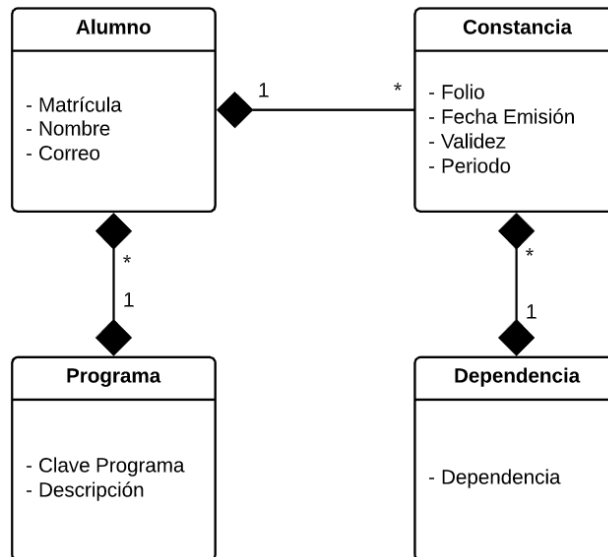


Figura 3.10: Modelo de dominio, módulo de validación de constancias de estudiante inscrito. (Elaboración propia).

Diagramas de casos de uso

El diagrama en la Figura 3.11, muestra la interacción del coordinador con el módulo de coordinadores, el cual tiene como objetivo ayudar al coordinador a subir su firma y sello de programa digitalizados, al sistema de emisión de constancias de estudiante inscrito, además de actualizar los registros de estudiantes inscritos en el programa que coordina.

En la Figura 3.12, se muestra la interacción del estudiante en el módulo de constancias de estudiante inscrito, en donde puede solicitar y descargar una constancia digital, además de consultar las constancias generadas con anterioridad, con opción de descargarlas nuevamente.

En la Figura 3.13, se describen los casos de uso para el módulo de validación de constancias de estudiante inscrito, donde la entidad validadora, puede confirmar la autenticidad de una constancia digital emitida por el prototipo de sistema, mediante el código de validación de la misma.

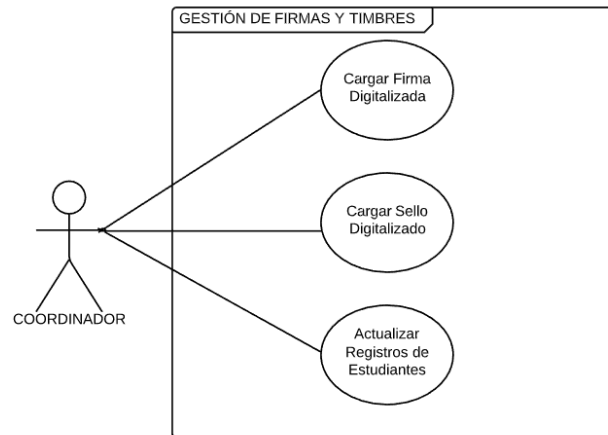


Figura 3.11: Diagrama de casos de uso para el módulo de coordinadores. (Elaboración propia).

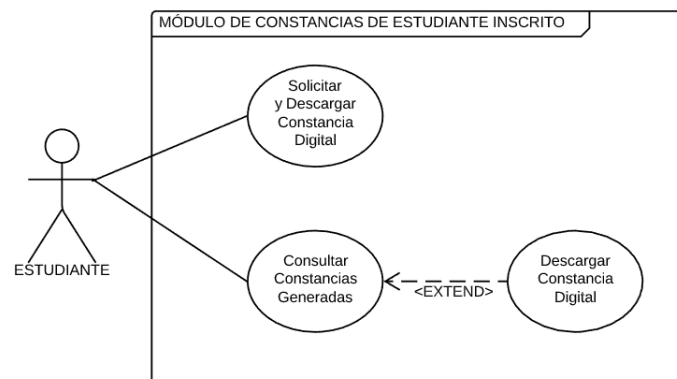


Figura 3.12: Diagrama de casos de uso para el módulo de constancias de estudiante inscrito. (Elaboración propia).

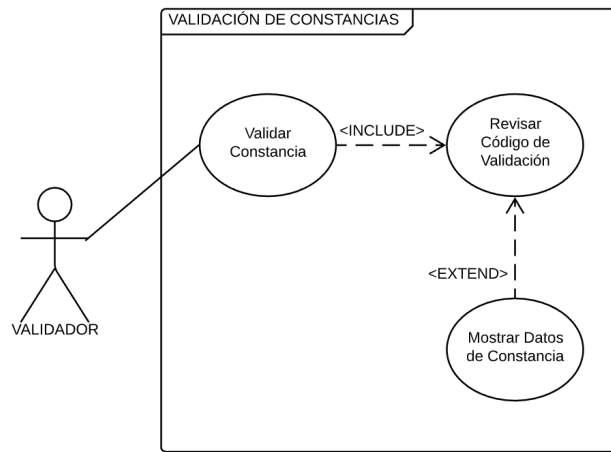


Figura 3.13: Diagrama de casos de uso para el módulo de validación de constancias de estudiante inscrito. (Elaboración propia).

3.2.4. Diseño del prototipo del sistema

Para el acceso a los módulos de coordinadores y el módulo de constancias, se elaboró una pantalla de inicio de sesión, utiliza el sistema de autenticación de *Laravel*, el cual provee un cifrado AES-256 para la gestión de contraseñas de usuario. A cada usuario registrado en el sistema, se le asigna un rol determinado, pudiendo ser coordinador, estudiante o administrador. La Figura 3.14, muestra el prototipo de pantalla para el inicio de sesión en el sistema. Utiliza el controlador *Login*, que se define más adelante.

La navegación entre los diversos módulos del prototipo de sistema, además de estar basada en el sistema de autenticación, se realizó con base en el rol del usuario, es así que, si un usuario no se encuentra registrado o no cuenta con un rol determinado, se redirigirá la vista al módulo de validaciones, a través de las funciones empleadas en los controladores de cada módulo, como se muestra en la Figura 3.15.

UACJ | UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Iniciar Sesión

Prototipo de Sistema de Constancias Electrónicas

Iniciar Sesión

Correo Electrónico Institucional

Contraseña

☐ Recordarme

Iniciar Sesión Olvidé mi contraseña

Av. Plutarco Elías Calles #1210 • Fovissste Chamizal • Ciudad Juárez, Chihuahua, México • C.P. 32310 • Tel. (+52) 688 – 2100 al 09

Figura 3.14: Prototipo de pantalla de inicio de sesión. (Elaboración propia).

3.2.5. Desarrollo del prototipo del sistema

Para el desarrollo del prototipo de sistema, se utilizó *MySQL Server 8.0*, como sistema gestor de bases de datos. Para el diseño de la base de datos, se empleó *MySQL Workbench* en su versión 6.3. En conjunto, estas herramientas permitieron el desarrollo de la base de datos del sistema. Además de otras funciones, como la creación de consultas, migraciones y respaldos.

Una vez que se implementó la base de datos a utilizar, se realizó la instalación del paquete XAMPP en su versión 7.2.10, para proveer un servidor web *Apache*, que permitió alojar los diferentes módulos del prototipo de sistema. Proveyendo un entorno seguro y confiable, para el desarrollo y pruebas de funcionamiento.

Se utilizó el *framework* de *Laravel* en su versión 5.8.3 en conjunto con el lenguaje PHP

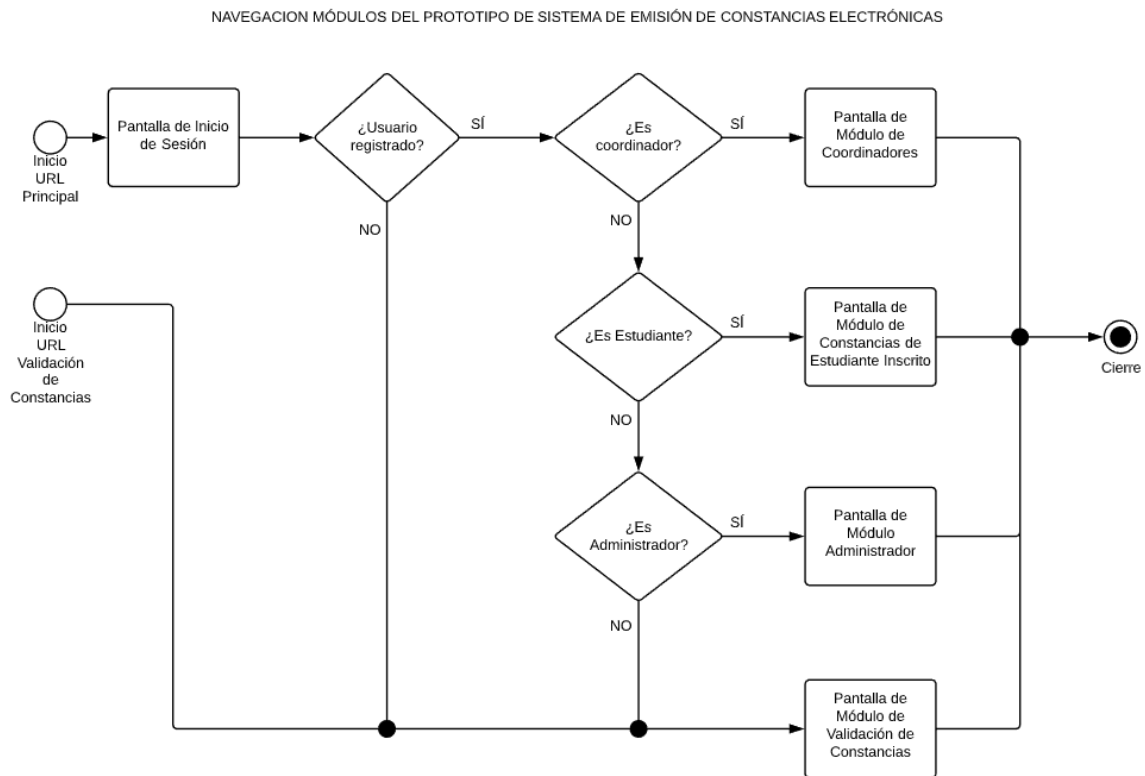


Figura 3.15: Diagrama de navegación entre módulos del prototipo de sistema. (Elaboración propia).

versión 7.2.10 para el desarrollo de modelos, *middleware* y controladores. Además de incorporar PHP con *VueJS* versión 2.5.17, para la creación de las vistas, aportando lógica a las mismas mediante el uso de *JavaScript*.

Controladores

Los controladores de los diversos módulos, se programaron en lenguaje PHP, utilizando funciones específicas para cada uno de los eventos particulares de los casos de uso que se generan en las vistas. Por ejemplo, en cada controlador se incluyó una función, que, en conjunto con el sistema de autenticación provisto por *Laravel*, redirecciona a la vista correspondiente asignada para un rol específico de usuario. Es así, que no solo se verifica que el usuario se haya autenticado como tal, sino que también, se asegura que solo podrá ingresar a la vista asignada a su rol desempeñado. Ver Figura 3.16.

```
public function index()
{
    if (Auth::check() && Auth::user()->role == 'coordinador') {
        return view('coordinadores');
    }
    elseif (Auth::check() && Auth::user()->role == 'alumno') {
        return view('constancias');
    }
    elseif (Auth::check() && Auth::user()->role == 'admin'){
        return view('admin');
    }
    else {
        return view('validacion');
    }
}
```

Figura 3.16: Función de redireccionamiento de vistas, incluida en controladores de módulos. (Elaboración propia).

En el controlador del módulo de validación, se incluyó una función para retornar los datos de la constancia digital que ha de ser validada, la cuál recibe como parámetro el código de validación de la constancia, codificado en *base64*. Cuando el código es recibido, esta función

verifica que se trate de una codificación válida, después, mediante el uso de expresiones regulares, se valida el formato y se descompone la cadena para extraer el folio y la matrícula asociada a la constancia. Una vez realizados estos pasos, se ejecuta la consulta a la base de datos. Ver Figura 3.17.

```
public function validarConstancia(Request $request){
    //Se decodifica el folio a partir del código de validación
    $folio = base64_decode($request->codigo, true);
    if(false === $folio) return 0;

    //Si no contiene guión para separar, se interrumpe el proceso
    if (!preg_match('/^[0-9]+[\-]+[0-9]/', $folio)) return 0;

    //Se extrae de la cadena decodificada el folio y la matrícula (fueron separados por '-')
    $folio_matricula = explode("-", $folio);
    $folio = $folio_matricula[0];
    $matricula = $folio_matricula[1];

    //Se utilizan expresiones regulares de PHP para la validación del formato de código
    //Si el folio o matrícula no es numérico, se interrumpe el proceso
    if (!preg_match('/^[0-9]*$/', $folio)) return 0;
    if (!preg_match('/^[0-9]*$/', $matricula)) return 0;

    //Se extraen los datos de la constancia que coincida con el código de validación
    //..... SQL QUERY.....
    $sql6 = "SELECT fechaEmision, fechaValidez, alumnos.matricula, alumnos.nombreAlumno as nombreAlumno, clavePrograma,
    descripcionPrograma, dependencia FROM constancias, alumnos, programas, dependencias
    WHERE dependencias_idDependencia = idDependencia AND alumnos.matricula = constancias.alumnos_matricula AND claveProgr
    AND folio = '". $folio ."' AND alumnos.matricula = $matricula
    AND fechaValidez >= curdate() AND alumnos_matricula = alumnos.matricula";

    $constancia = DB::select($sql6);

    return $constancia;
}
```

Figura 3.17: Función de validación de constancia, incluida en el controlador del módulo de validaciones. (Elaboración propia).

Vistas

Para el desarrollo de las vistas, se utilizaron los componentes de *VueJS*, lo que proporcionó reactividad en los diferentes elementos de los módulos. Por ejemplo, para el módulo de coordinadores, se utilizaron un total de siete componentes (ver Figura 3.18), cada uno de ellos para proveer una vista específica según el comportamiento del usuario.

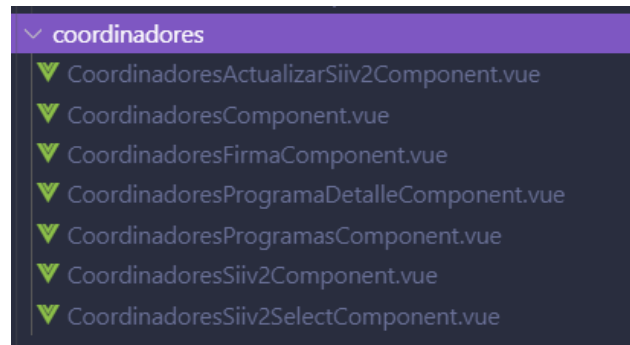


Figura 3.18: Componentes para el módulo de coordinadores. (Elaboración propia).

Constancia digital en PDF

Para la generación de la constancia digital, se utilizaron los paquetes *moment* versión 2.24.0, que proporcionó una adecuada gestión de los formatos de fecha y hora, además de *pdfMake* en su versión 0.1.58, para la generación del documento en formato PDF e imágenes codificadas en *base64*, que se utilizaron para generar el membrete institucional y el pie de página. En la Figura 3.19, se muestra el empleo de estas herramientas.

```
generarPDF(){
  this.preparaVariables()
  var docName = (this.constancia.constancia[0].fechaEmision).concat('.pdf')
  let sello = 'data:'.concat(this.constancia.sello[0].tipo).concat(';base64,').concat(this.constancia.sello[0].sello)
  let firma = 'data:'.concat(this.constancia.firma[0].tipo).concat(';base64,').concat(this.constancia.firma[0].firma)
  let fecha = moment(this.constancia.constancia[0].fechaEmision).locale('es').format('D [de] MMMM [de] YYYY')

  var pdfMake = require('pdfmake/build/pdfmake.js')
  if (pdfMake.vfs == undefined){
    var pdfFonts = require('pdfmake/build/vfs_fonts.js')
    pdfMake.vfs = pdfFonts.pdfMake.vfs;
  }

  var docDefinition = { ...
  }
  pdfMake.createPdf(docDefinition).download(docName)
}
```

Figura 3.19: Función para la generación de constancia digital en PDF. (Elaboración propia).

Además de los elementos mencionados en los requerimientos de la constancia, se incluyó la generación de un código QR, en él, se incrusta la URL de validación junto con el código de la constancia, lo que permite una validación directa. Este código se puede apreciar en el prototipo de constancia electrónica en la Figura 3.20.

Para la obtención de la constancia generada como prototipo, se utilizaron datos de prueba.

Instituto de Ingeniería y Tecnología

 UNIVERSIDAD AUTÓNOMA
DE CIUDAD JUÁREZ

Cd. Juárez, Chih., a 11 de septiembre de 2019

A quien corresponda
PRESENTE.

Se hace constar que **RIVAS FLORES CESAR ALONSO**, con número de matrícula **143468**, se encuentra **INSCRITO(A)** en el programa de **Ingeniería en Sistemas Computacionales**, actualmente cursa el **noveno semestre** en el periodo **Agosto - Diciembre 2019**, con un promedio general de 9.91, fecha inicio de semestre 5 de agosto de 2019, fecha de final de semestre 23 de noviembre de 2019, incluyendo periodo vacacional del 24 de noviembre de 2019 al 5 de enero de 2020. Clave institucional 08MSU0245B.

Por tal motivo no hay inconveniente alguno en extender la presente, para los fines legales que al(la) interesado(a) convengan.

Atentamente

**"Por una Vida Científica
Por una Ciencia Vital"**



Mtro. René Noriega Armendáriz
**Coord. del Programa de Ingeniería en Sistemas
Computacionales**



Código de Verificación: MzMtMTQzNDY4
Fecha/Hora de emisión: 11 de septiembre de 2019 22:40
Válido hasta: 5 de enero de 2020

Verifique este documento en: http://localhost/constancias/public/validacion_directa?codigo=MzMtMTQzNDY4

Figura 3.20: Prototipo de constancia electrónica en formato PDF. (Elaboración Propia).

Capítulo 4

Resultados y Discusiones

En este capítulo se presentan los resultados obtenidos después de haber desarrollado el proyecto, realizado pruebas de ejecución y desempeño, así como de haber elaborado entrevistas con usuarios directos de los distintos módulos de la aplicación. Además, se interpretan estos resultados de acuerdo a los objetivos planteados al inicio de esta investigación.

4.1. Presentación de resultados

Como resultado de la investigación, se desarrolló un prototipo de aplicación web que redujo en un 33 % los actores involucrados en el proceso manual que se realiza actualmente, puesto que, con su implementación, la secretaria del jefe de departamento dejaría de elaborar las constancias. Además, la participación del coordinador de programa puede ser reducida hasta en dos veces por año, esto para mantener actualizada tanto la base de datos de estudiantes inscritos, como el sello del programa y firma personal digitalizados, realizando estas tareas desde los submódulos dentro del módulo de coordinadores, como se muestra en la Figura 4.1.

El tiempo de respuesta hacia los estudiantes, para la obtención de una constancia que lo avale como estudiante inscrito, se redujo de dos días hábiles empleados en el proceso manual, a cuestión de segundos utilizando la aplicación web, esto tomando en cuenta que el nuevo formato propuesto es electrónico y no impreso, pero dándole oportunidad al interesado de

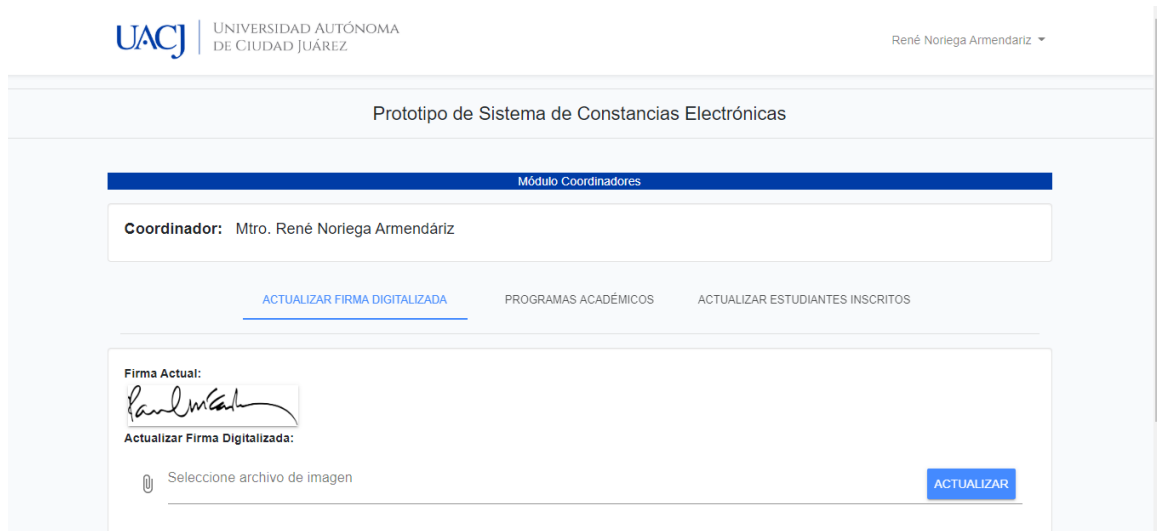


Figura 4.1: Pantalla principal del módulo de coordinadores. (Elaboración propia).

poder imprimir por su cuenta la constancia generada, sin que esta pierda validez, desde el módulo de estudiantes dentro de la aplicación (ver Figura 4.2).

La constancia electrónica de estudiante inscrito, como producto final, se genera con protección contra copia de elementos incluidos en la misma, así como un código de verificación impreso e incrustado en un código QR, para que pueda ser validada por cualquier entidad a través de Internet, dentro del módulo de validación de la aplicación web (ver Figura 4.3), ya sea desde dispositivos móviles o de escritorio, en este módulo no es necesario tener una sesión iniciada.

Para evaluar la funcionalidad del módulo de estudiantes, se realizaron pruebas con una muestra de 30 dispositivos de escritorio y laptops con 2 versiones de navegadores distintas, además de una muestra de 20 dispositivos móviles con 3 versiones diferentes de sistema operativo, como resultado, se logró acceder en el 100 % de las muestras, a las distintas funcionalidades del módulo (ver Figura 4.4).

Para medir el índice de usabilidad, se propuso una serie de tareas (ver Figura 4.5) a una

UACJ | UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

César Alonso Rivas Flores ▾

Prototipo de Sistema de Constancias Electrónicas

Módulo de Constancias de Estudiante Inscrito

Matricula 143468	Nombre RIVAS FLORES CESAR ALONSO	Prom. General 9.91
Clave de Programa 45200 - 11	Descripción del Programa Ingeniería en Sistemas Computacionales	Estatus INSCRITO

SOLICITUDES CONSULTAS

Selecciona destinatario:
A quien corresponda ▾

☐ Incluir Promedio General

Figura 4.2: Pantalla principal módulo de estudiante. (Elaboración propia).

UACJ | UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Iniciar Sesión

Prototipo de Sistema de Constancias Electrónicas

Módulo de Validación de Constancias de Estudiante Inscrito

Constancia Validada Satisfactoriamente

Datos de la Constancia

Matricula 143468	Nombre RIVAS FLORES CESAR ALONSO	Clave de Programa 45200 - 11	Descripción del Programa Ingeniería en Sistemas Computacionales
Dirigida a Secretaría del Trabajo y Previsión Social	Fecha/Hora de Emisión 2019-10-24 21:42:34	Válida Hasta 2020-01-12	

Figura 4.3: Pantalla de resultado de validación de constancia electrónica. (Elaboración propia).

Funcionalidad	Escritorio/Laptop		Dispositivo Móvil		
	Chrome v78.0	Firefox v69.0	Andorid 8.0	Android 7.0	Android 6.0
Inicio de Sesión	100%	100%	100%	100%	100%
Generación de constancia en la base de datos de la aplicación	100%	100%	100%	100%	100%
Descarga de constancia electrónica en formato PDF al dispositivo	100%	100%	100%	100%	100%
Consulta de constancias generadas	100%	100%	100%	100%	100%
Descarga de constancias consultadas	100%	100%	100%	100%	100%
Tamaño de la muestra:	30		20		

Figura 4.4: Porcentaje de funcionamiento módulo de estudiantes. (Elaboración propia).

muestra de 20 usuarios, al finalizar las tareas, se realizaron entrevistas directas, donde las preguntas se enfocaron en conocer factores específicos de la aplicación, tales como, la facilidad de aprendizaje, facilidad de eficiencia y uso, facilidad de recordar cómo funciona, frecuencia de errores y satisfacción subjetiva. Además de conocer en cuántos de los casos se completaron satisfactoriamente las tareas.

4.2. Interpretación de resultados

La Figura 4.5, muestra una gráfica con el porcentaje de finalización de tareas propuestas a una muestra de 20 usuarios, dentro de las tareas solicitadas, se incluyeron:

- Iniciar sesión en la aplicación web, donde se redirige automáticamente al módulo de estudiantes.
- Seleccionar una dependencia destino a la cual será dirigida la constancia.
- Descargar la constancia electrónica en formato PDF a su dispositivo.
- Consultar la constancia generada.
- Descargar la constancia consultada a su dispositivo.

Como se observa en la Figura 4.5, se obtuvo un índice muy alto de terminación de tareas, por lo que se deduce una interfaz intuitiva y de fácil uso.

La Figura 4.6, muestra el índice de satisfacción general subjetiva de los usuarios que realizaron las tareas propuestas, considerando aspectos como la dificultad percibida en la tarea dada, tiempo para completar la tarea, cualquier error cometido involuntariamente y la percepción general de la experiencia.

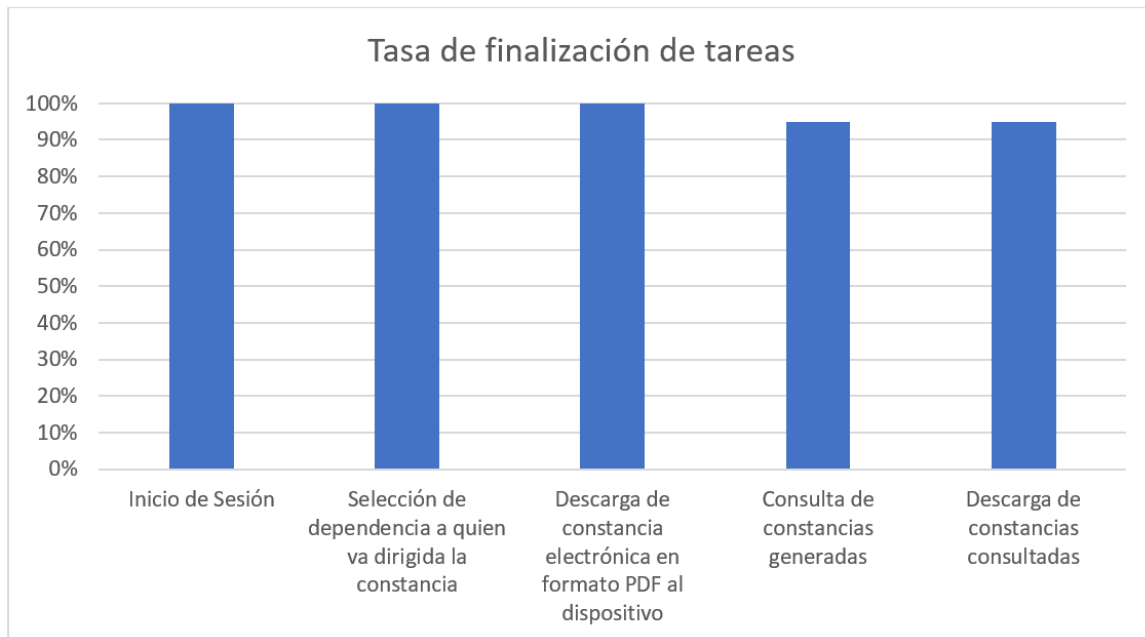


Figura 4.5: Tasa de finalización de tareas. (Elaboración propia).

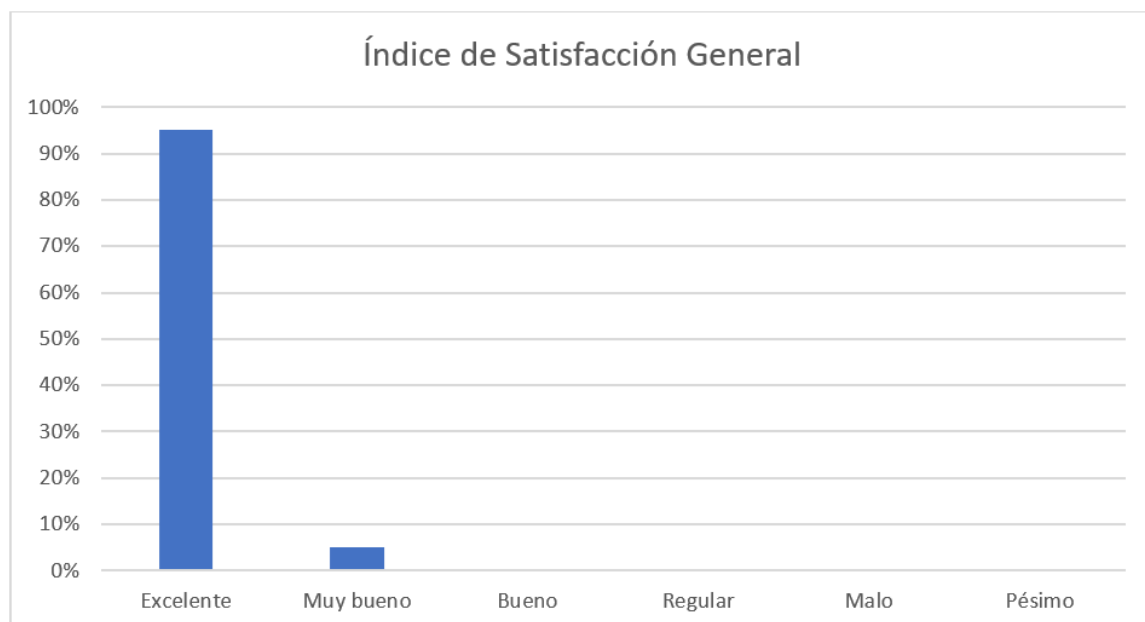


Figura 4.6: Índice de satisfacción general. (Elaboración propia).

Capítulo 5

Conclusiones

En este capítulo se abordan las conclusiones obtenidas con respecto al objetivo de la investigación, así como recomendaciones para futuras investigaciones.

5.1. Con respecto al objetivo de la investigación

Con base en los resultados obtenidos en la presente investigación, se brinda una solución específica a los problemas identificados de subutilización de recursos, tiempo invertido, falta de seguimiento, disponibilidad y error humano, que se desprenden de la ejecución manual del proceso de generación de constancias de estudiante inscrito en las diversas coordinaciones de los programas educativos en el IIT de la UACJ. Dicha solución fue generada a partir de un análisis metodológico, que reduce el impacto negativo al proceso, ocasionado por los problemas mencionados, y así, garantizar la eficiencia a través de la implementación de un sistema de información, en conjunto con el uso de las tecnologías disponibles en el entorno donde el proceso se desarrolla.

Con lo anterior se cumple el objetivo de esta investigación, logrando automatizar el proceso de emisión de constancias de estudiante inscrito, haciéndolo disponible para el estudiante desde cualquier lugar, 24 horas al día, además de contribuir a estandarizar el proceso para las distintas coordinaciones y departamentos.

5.2. Recomendaciones a futuro

- Para la obtención de los datos que alimentan la base de datos de la aplicación desarrollada, se empleó un proceso manual por parte de los coordinadores, quienes deben de generar un archivo CSV (valores separados por comas), desde el módulo de control escolar del *SiiV2*. Aunque este proceso debe realizarse como mínimo una vez por semestre, es importante tomar en cuenta cuando se desarrolla un sistema que se alimenta o necesita obtener datos de un sistema externo, como en este caso en particular, aspectos de interconexión entre los diferentes sistemas para reducir errores debido al factor humano.
- Es importante que la base de datos de la aplicación, se mantenga debidamente actualizada, tanto para brindar al estudiante inscrito la oportunidad de generar una constancia vigente, como para impedir que estudiantes dados de baja, puedan generar una constancia vigente.
- El cálculo de semestre en el que el estudiante se encuentra inscrito, dato que se muestra en la constancia generada por la aplicación web, se hizo con base en los créditos cursados y no por el semestre cronológico en el que el estudiante se encuentre. Es por ello que se hace la recomendación de profundizar en este punto, para establecer de una forma consensuada la manera en que el dato deberá calcularse, para este uso en particular.

Bibliografía

- [1] “Copia certificada de acta de nacimiento en línea”, *Gobierno Federal*, 2018. [En línea]. Disponible en: <https://www.gob.mx/actas>. [Accedido: 20/02/2019].
- [2] A. García and A. Morales, “Se desata el boom de actas de nacimiento en línea”, *El Universal*, 2018. [En línea]. Disponible en: <https://www.eluniversal.com.mx/nacion/sociedad/se-desata-boom-de-actas-de-nacimiento-en-linea>. [Accedido: 19/02/2019].
- [3] C. Ávila and M. Ramírez, “Análisis y diseño del sistema de control de servicio social de la uaeh”, tesis de pregrado, UAEH, 2005. [En línea]. Disponible en: <https://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Analisis%20y%20diseno%20del%20sistema.pdf>. [Accedido: 21/02/2019].
- [4] J. E. Zarazúa, “Aplicación para emisión de constancias electrónicas”, tesis de maestría, IPN, 2016. [En línea]. Disponible en: <http://148.204.210.201/tesis/1471374222967TESISJEZEMAES.pdf>. [Accedido: 21/02/2019].
- [5] A. C. Martínez, M. Gameros, J. C. Valdivia, L. P. Cervantes, R. Duarte, and Á. F. Gómez, “Anuario estadístico 2017-2018”, *Universidad Autónoma de Ciudad Juárez*, 2018. [En línea]. Disponible en: http://www.uacj.mx/DGPDI/Documents/SEI/Estad%C3%ADstica%20Institucional/Anuario_2017-2018.pdf. [Accedido: 20/02/2019].

- [6] A. Orozco, M. Gameros, J. C. Valdivia, and O. N. Fernández, “Agenda estadística 2018-2019”, *Universidad Autónoma de Ciudad Juárez*, 2019. [En línea]. Disponible en: <http://www.uacj.mx/DGPDI/SPCOA/Documents/estadisticas/ESTADISTICAS-2018-19%20%20A0.pdf>. [Accedido: 20/02/2019].
- [7] A. Vázquez, “Metodologías para el análisis y diseño de sistemas”, *Porlamar*, 2014. [En línea]. Disponible en: <https://bit.ly/2NkJWcp>. [Accedido: 22/02/2019].
- [8] J. Pacheco, “Qué es el modelado de procesos y cuáles son los beneficios para su empresa”, *Heflo*, 2017. [En línea]. Disponible en: <https://www.heflo.com/es/blog/mapeo-procesos/que-es-modelado-proceso/>. [Accedido: 22/03/2019].
- [9] J. Rumbaugh, I. Jacobson, and G. Booch, *El lenguaje unificado de modelado. Manual de referencia*, Madrid: Addison-Wesley, 2000.
- [10] M. Fowler, *UML gota a gota*, México: Addison-Wesley Longman de México. S.A. de C.V., 1999.
- [11] M. P. Usaola, “Desarrollo de software basado en reutilización”, *Universitat Oberta de Catalunya*, 2012. [En línea]. Disponible en: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingenieria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_\(Modulo_2\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingenieria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_(Modulo_2).pdf). [Accedido: 22/03/2019].
- [12] F. Durán, J. Troya, and A. Vallecillo, “Desarrollo de software dirigido por modelos”, *Fundación para la Universitat Oberta de Catalunya*, 2012. [En línea]. Disponible en: <https://pdfs.semanticscholar.org/cad3/e6dcc0a4cf9525a4372db94a030379e5ed68.pdf>. [Accedido: 22/03/2019].
- [13] “About the business process model and notation specification version 2.0”, *Object management group*, 2019. [En línea]. Disponible en: <https://www.omg.org/spec/BPMN/2.0/>. [Accedido: 26/03/2019].

- [14] M. Devment, “Como hacer un dfd – diagrama de flujo de datos”, *atic.cl*, 2013. [En línea]. Disponible en: <http://www.atic.cl/diagrama-de-flujo-de-datos-dfd/>. [Accedido: 29/03/2019].
- [15] R. S. Pressman, *Ingeniería del software un enfoque práctico*, México: McGraw-Hill, 2010.
- [16] C. Larman, *UML y Patrones*, México: Prentice Hall, 2003.
- [17] “¿qué son los prototipos de un sistema informático?”, *Alegsa*, 2010. [En línea]. Disponible en: <http://www.alegsa.com.ar/Diccionario/C/7664.php>. [Accedido: 29/03/2019].
- [18] I. Gilfillan, *La biblia MySQL*, México: Anaya Multimedia, 2003.
- [19] R. Camps, L. A. Casillas, D. Costal, M. Gibert, C. Martín, and O. Pérez, *Software libre bases de datos*, Catalunya: Fundació per a la Universitat Oberta de Catalunya, 2005.
- [20] E. Mercado, “Clasificación de bases de datos”, *Academia.edu*, 2015. [En línea]. Disponible en: https://www.academia.edu/14797747/Clasificacion_de_bases_de_datos. [Accedido: 05/04/2019].
- [21] L. Quintas, “Sistemas de gestión de bases de datos”, *Eoi escuela de negocios*, 2008. [En línea]. Disponible en: http://api.eoi.es/api_v1_dev.php/fedora/asset/eoi:45500/componente45499.pdf. [Accedido: 06/04/2019].
- [22] Z. Tejada, “Datos no relacionales y nosql”, *Microsoft*, 2018. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>. [Accedido: 05/04/2019].
- [23] “¿qué es nosql?”, *Amazon Web Services*, 2019. [En línea]. Disponible en: <https://aws.amazon.com/es/nosql/>. [Accedido: 06/04/2019].

- [24] S. Sæther, A. Aulbach, E. Schmid, J. Winstead, L. Torben, R. Lerdorf, A. Zmievski, J. Ahto, R. Martínez, V. Fernández, and L. Boshell, *Manual de PHP*, CA, USA: Zend Technologies Ltd, 2002.
- [25] A. Muñoz, *Introducción a Node.JS a través de Koans*, Madrid: Arturo Muñoz de la Torre, 2013.
- [26] J. D. Gauchat, *El gran libro de HTML5, CSS3 y Javascript*, Barcelona: Marcombo, 2012.
- [27] J. Pavón, “Estructura de las aplicaciones orientadas a objetos el patrón modelo-vista-controlador (mvc)”, *Universidad Complutense Madrid*, 2009. [En línea]. Disponible en: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>. [Accedido: 06/04/2019].
- [28] Y. Fernández and Y. Díaz, “Patrón modelo-vista-controlador”, *Telem@tica*, vol. 11, no. 1, pp. 47–57, 2012.
- [29] A. Fontela, “Qué es bootstrap”, *Raiola Networks*, 2015. [En línea]. Disponible en: <https://raiolanetworks.es/blog/que-es-bootstrap/>. [Accedido: 05/04/2019].
- [30] G. Pantaleo and L. Rinaudo, *Ingeniería de Software*, Buenos Aires: Alfaomega, 2015.
- [31] A. Weitzenfeld and S. Guardati, “Ingeniería de software: el proceso para el desarrollo de software”, *Robolat*, 2007. [En línea]. Disponible en: <http://weitzenfeld.robolat.org/wp-content/uploads/2015/01/WeitzenfeldGuardatiComputacion2008.pdf>. [Accedido: 06/04/2019].
- [32] A. Martínez and R. Martínez, “Guía a rational unified process”, *Universidad de Castilla la Mancha*, 2011. [En línea]. Disponible en: <https://anaylenlopez.files.wordpress.com/2011/03/trabajo-guia20rup.pdf>. [Accedido: 29/03/2019].

- [33] “Manual de metodologías Ágiles”, *Wingu*, 2016. [En línea]. Disponible en: https://www.winguweb.org/system/files/biblioteca/manual_de_metologias_agiles_final.pdf. [Accedido: 06/04/2019].
- [34] A. Peralta, “Metodología scrum”, *Universidad ORT Uruguay*, 2003. [En línea]. Disponible en: <https://fi.ort.edu.uy/innovaportal/file/2021/1/scrums.pdf>. [Accedido: 06/04/2019].
- [35] A. Medina and J. Navarro, “Diseño de un sistema para optimizar el proceso de inscripciones de la unidad educativa fundación educativa maranatha, ubicada en el municipio san diego del estado carabobo”, tesis de pregrado, Universidad de Carabobo, 2015. [En línea]. Disponible en: <http://mriuc.bc.uc.edu.ve/handle/123456789/2227>. [Accedido: 20/02/2019].
- [36] M. Frayssinet, “Taller de implementación de la norma iso 27001”, *onpei.gob.pe*, 2011. [En línea]. Disponible en: https://www.gobiernodigital.gob.pe/docs/ISO_27001_v011.pdf. [Accedido: 06/04/2019].
- [37] D. M. Terán, *Administración estratégica de la función informática*, México: Alfaomega, 2014.
- [38] “Cortafuegos”, *IBM Knowledge Center*, 2017. [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_71/rzaj4/rzaj4fwfirewallconcept.htm. [Accedido: 07/04/2019].
- [39] “El portal de iso 27001 en español”, *iso27000.es*, 2005. [En línea]. Disponible en: <http://www.iso27000.es/iso27000.html>. [Accedido: 06/04/2019].
- [40] L. Perurena and M. Moráquez, “Usabilidad de los sitios web, los métodos y las técnicas para la evaluación”, *Revista cubana de información en ciencias de la salud*, vol. 24, no. 2, pp. 1–16, 2013.

- [41] A. B. Martínez and J. M. Cueva, “Estándares y guías”, *Universidad de Oviedo*, 2001. [En línea]. Disponible en: <https://aipo.es/libro/pdf/09Estand.pdf>. [Accedido: 06/04/2019].

Formato de solicitud de constancias departamento de física y matemáticas

DMA

79

Apéndice B

Estructura del proyecto

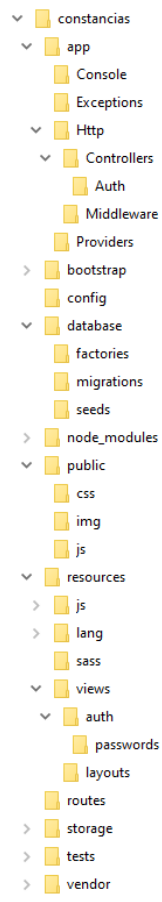


Figura B.1: Estructura de directorios del sistema.

Apéndice C

Funciones principales

```
public function actualizarSiiv2(Request $request){
    $tipo = mime_content_type(utf8_encode($request->file));
    $archivotmp = file_get_contents($request->file);
    $ruta_tmp = request()->file('file')->path();
    $bad_chars = "\\\"";
    $ruta = str_replace($bad_chars, '/', $ruta_tmp);

    $sql = DB::connection()->getPdo()
        ->exec("
        LOAD DATA LOCAL INFILE '$ruta.'"
        INTO TABLE siiv2
        FIELDS TERMINATED BY ','
        LINES TERMINATED BY '\n'
        IGNORE 1 ROWS;");

    return $sql;
}
```

Figura C.1: Función para actualizar la tabla de apoyo Siiv2.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;

class AdminController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('admin');
    }

    public function index()
    {
        if (Auth::check() && Auth::user()->role == 'coordinador') {
            return view('coordinadores');
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return view('constancias');
        }
        elseif (Auth::check() && Auth::user()->role == 'admin'){
            return view('admin');
        }
        else {
            return view('validacion');
        }
    }
}
```

Figura C.2: Función principal del módulo de administrador.

```
public function bajarFirma(){
    $correo = Auth::user()->email;
    $sql = "SELECT idCoordinador
            FROM coordinadores WHERE correoCoordinador = '". $correo. "'";
    $idCoordinador = DB::select($sql);
    $id = (int) $idCoordinador;

    $sql2 = "SELECT firma, tipo FROM firmas WHERE coordinadores_idCoordinador = ?";
    $firma = DB::select($sql2, array($id));
    return $firma;
}
```

Figura C.3: Función para extraer de la B.D. la firma de un coordinador.

```
public function bajarSello($clavePrograma){

    $sql2 = "SELECT sello, tipo FROM sellos WHERE programas_clavePrograma = ?";
    $sello = DB::select($sql2, array($clavePrograma));
    return $sello;
}
```

Figura C.4: Función para extraer de la B.D. el sello de una coordinación.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;
use DB;

class ConstanciasController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('alumno');
    }

    public function index()
    {
        if (Auth::check() && Auth::user()->role == 'coordinador') {
            return view('coordinadores');
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return view('constancias');
        }
        elseif (Auth::check() && Auth::user()->role == 'admin'){
            return view('admin');
        }
        else {
            return view('validacion');
        }
    }
}
```

Figura C.5: Función principal del módulo de constancias de estudiante inscrito.

```
public function consultaConstancias(Request $request){  
    //Función para devolver el listado de constancias vigentes para la matrícula dada  
    $matricula = $request->matricula;  
  
    $sql = "SELECT folio, alumnos_matricula, fechaEmision, fechaValidez, dependencia FROM constancias, alumnos, dependencias  
    WHERE matricula=alumnos_matricula AND constancias.dependencias_idDependencia = idDependencia AND matricula = ' ".$matricula."  
    $constancias = DB::select($sql);  
  
    return $constancias;  
}
```

Figura C.6: Función para consultar las constancias generadas por un estudiante.

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Auth;

class LoginController extends Controller
{
    use AuthenticatesUsers;

    protected function redirectTo( ) {
        if (Auth::check() && Auth::user()->role == 'coordinador') {
            return('/coordinadores');
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return('/constancias');
        }
        elseif (Auth::check() && Auth::user()->role == 'admin'){
            return ('/admin');
        }
        else {
            return ('/validacion');
        }
    }

    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }
}
```

Figura C.7: Controlador de inicio de sesión en el sistema.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;
use DB;

class CoordinadoresController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('coordinador');
    }

    public function index()
    {
        if (Auth::check() && Auth::user()->role == 'coordinador') {
            return view('coordinadores');
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return view('constancias');
        }
        elseif (Auth::check() && Auth::user()->role == 'admin'){
            return view('admin');
        }
        else {
            return view('validacion');
        }
    }
}
```

Figura C.8: Función principal del módulo de coordinadores.

```
public function eliminarSiiv2(Request $request){
    $cvePrograma = $request->cvePrograma;

    $eliminar = DB::table('siiv2')
        ->where('cvePrograma', $cvePrograma)
        ->delete();

    return $eliminar;
}
```

Figura C.9: Función para eliminar registros de un determinado programa educativo.

```
generarPDF(){
  this.preparaVariables()
  var docName = (this.constancia.constancia[0].fechaEmision).concat('.pdf')
  let sello = 'data:'.concat(this.constancia.sello[0].tipo).concat(';base64,').concat(
    (this.constancia.sello[0].sello)
  )
  let firma = 'data:'.concat(this.constancia.firma[0].tipo).concat(';base64,').concat(
    (this.constancia.firma[0].firma)
  )
  let fecha = moment(this.constancia.constancia[0].fechaEmision).locale('es').format('D [de]
    MMMM [de] YYYY')

  var pdfMake = require('pdfmake/build/pdfmake.js')
  if (pdfMake.vfs == undefined){
    var pdfFonts = require('pdfmake/build/vfs_fonts.js')
    pdfMake.vfs = pdfFonts.pdfMake.vfs;
  }

  var docDefinition = {...
  }
  pdfMake.createPdf(docDefinition).download(docName)

  console.log("generada")
  console.log(this.constancia.constancia[0].nombreAlumno)
  console.log(this.constancia.opcionPromedio)
  console.log(this.constancia.vacaciones[0].fechaInicioVacaciones)
}
```

Figura C.10: Función para la generación de la constancia digital en formato PDF.

```

public function generarConstancia(Request $request){
    //variables para utilizar en la generación de la constancia
    $updated_at = (date("Y-m-d H:i:s", time()));
    $created_at = $updated_at;
    $fechaEmision = $updated_at;
    $matricula = $request->matricula;
    $idDependencia = $request->idDependencia;
    $opcionPromedio = $request->opcionPromedio;

    //Se revisa si existe una constancia generada por el alumno para la misma dependencia que aún esté vigente
    $sql = "SELECT folio FROM constancias WHERE alumnos_matricula = '".$matricula."'
        AND dependencias_idDependencia = '".$idDependencia."' AND fechaValidez >= curdate()";
    $existeConstancia = DB::select($sql);
    //Si existe una constancia vigente para la misma dependencia, se actualiza la fecha y se devuelve el registro encontrado
    if($existeConstancia != null){
        $folio = $existeConstancia[0]->folio;
        DB::table('constancias')
            ->where('folio', $folio)
            ->update(array('fechaEmision' => $fechaEmision, 'updated_at' => $updated_at));

        //Se extraen los datos de la constancia que coincida con el FOLIO que se actualizó y la MATRÍCULA del estudiante
        $sql6 = "SELECT folio, fechaEmision, fechaValidez, alumnos.matricula, nombreAlumno as nombreAlumno, clavePrograma,
            descripcionPrograma, CONCAT(titulo, nombreCoordinador) as nombreCoordinador,
            dependencia, promGral, creditosOblActuales, creditosOptActuales FROM constancias, periodos, alumnos, programas, coord
            WHERE dependencias_idDependencia = idDependencia AND alumnos.matricula = '".$matricula."' AND clavePrograma = program
            AND folio = '".$folio."' AND coordinadores_idCoordinador = idCoordinador
            AND fechaValidez >= curdate() AND siiv2.matricula = alumnos.matricula AND alumnos_matricula = alumnos.matricula";

        $constancia = DB::select($sql6);

        //Se extraen los datos del periodo, si hay más de uno activo, se extraen los del más reciente
        $sql4 = "SELECT cvePeriodo, descripcionPeriodo, fechaInicioPeriodo, fechaFinPeriodo FROM periodos
            WHERE estatusPeriodo='ACTIVO' ORDER by fechaInicioPeriodo DESC LIMIT 1";
        $periodo = DB::select($sql4);

        //Se extraen los datos de las vacaciones correspondiente al periodo anteriormente extraído
        $sql5 = "SELECT fechaInicioVacaciones, fechaFinVacaciones FROM vacaciones

```

Figura C.11: Función para la generación de la constancia digital.

```

public function listaProgramas(){
    $correo = Auth::user()->email;
    $sql = "SELECT idCoordinador
        FROM coordinadores WHERE correoCoordinador = '".$correo."'";
    $idCoordinador = DB::select($sql);
    $id = (int) $idCoordinador;

    $sql2 = "SELECT clavePrograma, descripcionPrograma, idSellos, sello FROM programas, sellos WHERE coordinadores_idCoordinac
    $programas = DB::select($sql2, array($id));
    return $programas;
}

```

Figura C.12: Función extraer de la B.D. los programas educativos de un coordinador en específico.

```
<?php

namespace App\Http\Middleware;

use Closure;
use Auth;

class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (Auth::check() && Auth::user()->role == 'admin') {
            return $next($request);
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return redirect('/constancias');
        }
        else {
            return redirect('/coordinadores');
        }
    }
}
```

Figura C.13: Middleware para el módulo de administrador.

```
<?php

namespace App\Http\Middleware;

use Closure;
use Auth;

class Alumno
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (Auth::check() && Auth::user()->role == 'alumno') {
            return $next($request);
        }
        elseif (Auth::check() && Auth::user()->role == 'coordinador') {
            return redirect('/coordinadores');
        }
        else {
            return redirect('/admin');
        }
    }
}
```

Figura C.14: Middleware para el módulo de estudiantes.

```
<?php

namespace App\Http\Middleware;

use Closure;
use Auth;

class Coordinador
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (Auth::check() && Auth::user()->role == 'coordinador') {
            return $next($request);
        }
        elseif (Auth::check() && Auth::user()->role == 'alumno') {
            return redirect('/constancias');
        }
        else {
            return redirect('/admin');
        }
    }
}
```

Figura C.15: Middleware para el módulo de coordinadores.

```

public function reimprimirConstancia(Request $request){
    $matricula = $request->matricula;
    $folio = $request->folio;
    $opcionPromedio = $request->opcionPromedio;

    //Se extraen los datos de la constancia que coincida con el FOLIO que se actualizó y la MATRÍCULA del estudiante
    $sql6 = "SELECT folio, fechaEmision, fechaValidez, alumnos.matricula, nombreAlumno as nombreAlumno, clavePrograma,
    descripcionPrograma, CONCAT(titulo, nombreCoordinador) as nombreCoordinador,
    dependencia, promGral, creditosOblActuales, creditosOptActuales FROM constancias, periodos, alumnos, programas, coordinadores
    WHERE dependencias_idDependencia = idDependencia AND alumnos.matricula = ".$matricula." AND clavePrograma = programas_clavePrograma
    AND folio = ".$folio." AND coordinadores_idCoordinador = idCoordinador
    AND fechaValidez >= curdate() AND siiv2.matricula = alumnos.matricula AND alumnos_matricula = alumnos.matricula";

    $constancia = DB::select($sql6);

    //Se extraen los datos del periodo, si hay más de uno activo, se extraen los del más reciente
    $sql4 = "SELECT cvePeriodo, descripcionPeriodo, fechaInicioPeriodo, fechaFinPeriodo FROM periodos
    WHERE estatusPeriodo='ACTIVO' ORDER by fechaInicioPeriodo DESC LIMIT 1";
    $periodo = DB::select($sql4);

    //Se extraen los datos de las vacaciones correspondiente al periodo anteriormente extraído
    $sql5 = "SELECT fechaInicioVacaciones, fechaFinVacaciones FROM vacaciones
    WHERE vacaciones.periodos_cvePeriodo = ?";
    $vacaciones = DB::select($sql5, array($periodo[0]->cvePeriodo));

    //Se extrae el sello de la coordinación
    $sql7 = "SELECT sello, tipo FROM sellos, alumnos, programas, coordinadores
    WHERE sellos.programas_clavePrograma = clavePrograma AND programas.coordinadores_idCoordinador = idCoordinador
    AND alumnos.programas_clavePrograma = clavePrograma AND alumnos.matricula = ".$matricula." ";
    $sello = DB::select($sql7);

    //Se extrae la firma del coordinador
    $sql8 = "SELECT firma, tipo FROM firmas, alumnos, programas, coordinadores
    WHERE firmas.coordinadores_idCoordinador = idCoordinador AND programas.coordinadores_idCoordinador = idCoordinador
    AND alumnos.programas_clavePrograma = clavePrograma AND alumnos.matricula = ".$matricula." ";
    $firma = DB::select($sql8);
}

```

Figura C.16: Función para la reimpresión de constancias.

```

public function revisarSemestre($semestre, $creditos){

    if(($semestre[0]->primero >0 && $creditos<=$semestre[0]->primero) || $semestre[0]->segundo == 0) {
        return "primer semestre";
    }elseif(($semestre[0]->segundo >0 && $creditos<=$semestre[0]->segundo) || $semestre[0]->tercero ==0){
        return "segundo semestre";
    }elseif(($semestre[0]->tercero >0 && $creditos<=$semestre[0]->tercero) || $semestre[0]->cuarto ==0){
        return "tercer semestre";
    }elseif(($semestre[0]->cuarto >0 && $creditos<=$semestre[0]->cuarto ) || $semestre[0]->quinto ==0){
        return "cuarto semestre";
    }elseif(($semestre[0]->quinto >0 && $creditos<=$semestre[0]->quinto) || $semestre[0]->sexto ==0){
        return "quinto semestre";
    }elseif(($semestre[0]->sexto >0 && $creditos<=$semestre[0]->sexto) || $semestre[0]->septimo ==0){
        return "sexto semestre";
    }elseif(($semestre[0]->septimo >0 && $creditos<=$semestre[0]->septimo) || $semestre[0]->octavo ==0){
        return "séptimo semestre";
    }elseif(($semestre[0]->octavo >0 && $creditos<=$semestre[0]->octavo) || $semestre[0]->noveno ==0){
        return "octavo semestre";
    }elseif(($semestre[0]->noveno >0 && $creditos<=$semestre[0]->noveno) || $semestre[0]->decimo ==0){
        return "noveno semestre";
    }elseif($semestre[0]->decimo >0 && $creditos<=$semestre[0]->decimo){
        return "décimo semestre";
    }elseif($semestre[0]->decimo >0 && $creditos>$semestre[0]->decimo){
        return "décimo semestre";
    }else{
        return "(no se determinó semestre)";
    }
}

```

Figura C.17: Función para calcular el semestre en el que un estudiante se encuentre inscrito con base en los créditos acreditados y cursados.

```

public function subirFirma(Request $request)
{
    $correo = Auth::user()->email;
    $sql = "SELECT idCoordinador
            FROM coordinadores WHERE correoCoordinador = '".$correo."'";
    $idCoordinador = DB::select($sql);
    $id = (int) $idCoordinador;

    $tipo = mime_content_type(utf8_encode($request->file));

    $image = file_get_contents($request->file);

    $response = DB::table('firmas')
    ->updateOrInsert(['coordinadores_idCoordinador' => $id],['firma' => base64_encode($image), 'tipo' => $tipo]);

    return (int)$response;
}

```

Figura C.18: Función para actualizar en la B.D. la firma digitalizada del coordinador.

```

public function subirSello(Request $request)
{
    $tipo = mime_content_type(utf8_encode($request->file));

    $image = file_get_contents($request->file);
    $clavePrograma = $request->clavePrograma;

    $response = DB::table('sellos')
    ->updateOrInsert(['programas_clavePrograma' => $clavePrograma], ['sello' => base64_encode($image), 'tipo' => $tipo]);

    return (int)$response;
}

```

Figura C.19: Función para actualizar en la B.D. el sello digitalizado de las coordinaciones.

```

public function validarConstancia(Request $request){
    //Se decodifica el folio a partir del código de validación
    $folio = base64_decode($request->codigo, true);
    if(false === $folio) return 0;

    //Si no contiene guión para separar, se interrumpe el proceso
    if (!preg_match('/^[0-9]+[-]+[0-9]/', $folio)) return 0;

    //Se extrae de la cadena decodificada el folio y la matrícula (fueron separados por '-')
    $folio_matricula = explode("-", $folio);
    $folio = $folio_matricula[0];
    $matricula = $folio_matricula[1];

    //Se utilizan expresiones regulares de PHP para la validación del formato de código
    //Si el folio o matrícula no es numérico, se interrumpe el proceso
    if (!preg_match('/^[0-9]*$/', $folio)) return 0;
    if (!preg_match('/^[0-9]*$/', $matricula)) return 0;

    //Se extraen los datos de la constancia que coincida con el código de validación
    //..... SQL QUERY.....
    $sql6 = "SELECT fechaEmision, fechaValidez, alumnos.matricula, alumnos.nombreAlumno as nombreAlumno, clavePrograma,
    descripcionPrograma, dependencia FROM constancias, alumnos, programas, dependencias
    WHERE dependencias_idDependencia = idDependencia AND alumnos.matricula = constancias.alumnos_matricula AND clavePrograma =
    AND folio = '".$folio."' AND alumnos.matricula = $matricula
    AND fechaValidez >= curdate() AND alumnos_matricula = alumnos.matricula";

    $constancia = DB::select($sql6);

    return $constancia;
}

```

Figura C.20: Función para validar una constancia generada por el sistema.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use DB;
use Exception;

class ValidacionController extends Controller
{
    public function index()
    {
        //return view('validacion');
        return view('validacion', array('codigo' => '0'));
    }

    public function validarCodigo(Request $codigoValidacion){

        return view('validacion_directa', array('codigo' => $codigoValidacion->codigo));
    }
}
```

Figura C.21: Función principal del módulo de validación.