



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

PROCESO DE TITULACIÓN

DICIEMBRE 2019 – SEPTIEMBRE 2020

PROPUESTA TECNOLÓGICA

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

TEMA

**SOFTWARE DE SIMULACION DEL EXAMEN COMPLEXIVO PARA EVALUAR LOS
CONOCIMIENTOS DE LOS ESTUDIANTES DE LA FACULTAD DE
ADMINISTRACION FINANZAS E INFORMATICA DE LA UTB**

EGRESADO

JORGE ALBERTO MORANTE JIMÉNEZ

TUTOR

ING. JOSÉ TEODORO MEJÍA VITERI.

2020

RESUMEN

Conforme a lo establecido por la Ley Orgánica De Educación Superior “LOES”, se deberá evaluar a los estudiantes de último año previo a la obtención del título respectivo, por lo que la carrera de Ingeniería de Sistema junto con la unidad de titulación, ofrecen opciones de titulación a los estudiantes de las cuales una de ellas es el examen Complexivo.

Para esta opción de titulación, la Universidad Técnica de Babahoyo no cuenta con un sistema web aplicativo que permita gestionar de manera óptima para la toma de los exámenes complexivos. Cabe recalcar que el manejo de información manual y el registro en papel, conlleva varios problemas como: la desorganización, manipulación inadecuada y posible pérdida de documentación sensible, por lo cual se ofreció una aplicación web que permita realizar de manera óptima la gestión correspondiente a utilizar en los exámenes complexivos considerando también los requerimientos relacionados a este proceso.

La aplicación en que se encuentra desarrollada a través del uso de Open Source tales como: SQLPostgres, Wampserver, NotePad++,PHP. Adicionalmente, se utilizó un patrón de arquitectura de software MVC (Modelo, Vista, Controlador), el cual separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el modelo encargado de gestionar los eventos y las comunicaciones, permitiendo facilidad de mantenimiento, reutilización del código y separación de conceptos.

El resultado final fue una aplicación web de fácil manejo y administración, que abarca los procesos involucrados y utilizados en el examen complexivos, desde la creación, seguimiento y aprobación de los mismos hasta la generación del examen de manera automática, permitiendo así, realizar de manera eficiente cada una de las tareas de la Unidad de Titulación.

Además, se brinda la opción de un simulador del examen, que estará disponible desde la página web de la unidad de titulación.

PALABRAS CLAVES:

- Open Source.
- Modelo Estructurado
- Lógica de Negocio
- Gestionar Eventos
- Ajax

ABSTRACT

In accordance with the provisions of the Organic Law of Higher Education "LOES", final year students must be evaluated prior to obtaining the respective title, so the System Engineering career together with the degree unit, offer options of qualification to the students of which one of them is the Complexive exam.

For this degree option, the Technical University of Babahoyo does not have a web application system that allows optimal management of complex exams. It should be noted that the handling of manual information and the paper record entails several problems such as: disorganization, improper handling and possible loss of sensitive documentation, for which a web application was offered that allows to optimally carry out the corresponding management to be used. in complex exams also considering the requirements related to this process.

The application in which it is developed through the use of Open Source such as: SQLPostgres, Wampserver, NotePad ++, PHP. Additionally, an MVC software architecture pattern (Model, View, Controller) was used, which separates the data and business logic of an application from the user interface and the model in charge of managing events and communications, allowing ease of maintenance, code reuse and separation of concepts.

The final result was a web application that is easy to use and administer, which covers the processes involved and used in the complex exam, from the creation, follow-up and approval of the same to the generation of the exam automatically, thus allowing them to be performed in an automatic way. efficient each of the tasks of the Degree Unit. In addition, the option of an exam simulator is offered, which will be available from the website of the degree unit.

KEYWORDS:

- Open Source.
- Structured Model
- Business logic
- Manage Events
- Ajax

INTRODUCCIÓN

La presente investigación tiene que ver con la elaboración y desarrollo de un sistema web informático, para **EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTE DE LA FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICA DE LA UTB**, este sistema web va a manejar información sobre los temas relacionados de todas las asignaturas desde el primer nivel hasta el último nivel de la carrera que se esté cursando.

Desde el principio de siglo las persona han buscado mejores formas de relacionar conceptos y medir su capacidad intelectual por medio de sistemas web informáticos, motivos por el cual y teniendo en cuenta las necesidades planteadas para para los estudiantes de la **FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICAS UTB**, se presenta la realización del **SOFTWARE DE SIMULACION DEL EXAMEN COMPLEXIVO PARA EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTES DE LA FACULTAD DE AMINISTRACION FINANZAS E INFORMATICA DE LA UTB**, este permite tener un control en la información que es de vital importancia para la facultad en la **ESCUELA DE SISTEMA**, ya que brinda a estudiantes conocer en qué nivel de educación y conceptualización de información se encuentra.

En el **SOFTWARE DE SIMULACION DEL EXAMEN COMPLEXIVO**, se reconoce como un software liviano y ajustable a las necesidades del usuario, brindando seguridad y facilidad en el manejo de la información rasgos que son importantes para cualquier institución educativa de hoy en día.

Igualmente, y teniendo en cuenta el rápido avance que tienen los sistemas de información en la actualidad, el sistema tiene como principal características que esta accesible desde Internet, plataforma mundialmente conocida y que ha brindado la apertura del mundo en grandes niveles,

dando entonces a la **FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICAS**
UTB una posibilidad más de reconocimiento por el trabajo realizado por sus estudiantes en proceso
de titulación en miras de dar apertura en el mercado institucional.

CAPITULO I

1. Diagnóstico de Necesidades y Requerimientos.

Conforme a lo establecido por la Ley Orgánica De Educación Superior “LOES”, se deberá evaluar a los estudiantes de último año previo a la obtención del título respectivo, por lo que la carrera de Ingeniería de Sistema junto con la unidad de titulación, ofrecen opciones de titulación a los estudiantes de las cuales una de ellas es el examen complexivos.

Para esta opción de titulación, la Universidad Técnica de Babahoyo no cuenta con un sistema web aplicativo que permita gestionar de manera óptima para la toma de los exámenes complexivos. Cabe recalcar que el manejo de información manual y du registro en papel, conlleva varios problemas como: la desorganización, manipulación inadecuada y posible pérdida de documentación sensible, por lo cual se ofreció una aplicación web que permita realizar de manera óptima la gestión correspondiente a utilizar en los exámenes complexivos considerando también los requerimientos relacionados a este proceso.

La aplicación en que se encuentra desarrollada a través del uso de Open Source tales como: SQLPostgres, Wampserver, NotePad++, MySql. Adicionalmente, se utilizó un patrón de arquitectura de software MVC (Modelo, Vista, Controlador), el cual separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el modelo encargado de gestionar los eventos y las comunicaciones, permitiendo facilidad de mantenimiento, reutilización del código y separación de conceptos.

El resultado final fue una aplicación web de fácil manejo y administración, que abarca los procesos involucrados y utilizados en el examen complexivos, desde la creación, seguimiento y aprobación de los mismos hasta la generación del examen de manera automática,

permitiendo así, realizar de manera eficiente cada una de las tareas de la Unidad de Titulación. Además, se brinda la opción de un simulador del examen, que estará disponible desde la página web de la unidad de titulación.

1.1.Ámbito de Aplicación: Descripción Del Contexto Y Hecho De Interés.

Teniendo en cuenta la necesidad que los estudiantes en formación de titulación de las diferentes carreras tantos profesionales, técnicas o estudios superiores como especialización, postgrados, maestrías, doctorados entre otras puedan evaluar sus capacidades de aprendizajes y actitudes, se plantea mejorar, proveer y desarrollar el conocimiento nivel de aprendizaje que es manejado por cada persona, esto basado en la resolución de problemas y entendimiento de los mismo. Igualmente, al estar el estudiante preparado para este tipo de pruebas de conocimientos especializados en ciertas áreas, se conocerá por su parte, en qué nivel se encuentra la carrera en cuestión y así se podrá tomar medidas en el nivel académico exigido dentro de la institución o centro educativo.

1.2.Establecimiento de Requerimientos.

Para la realización del sistema se necesitan los siguientes requerimientos tecnológicos:

HARDWARE: Característica del CPU.

- Procesador mínimo Core i3 de cuarta generación, máximo Core i7 de cuarta generación en adelante.
- Memoria RAM de 4 gb mínimo, máximo 8gb RAM.
- Hdd 500 gb mínimo, máximo 1tb.

Comentado [MOU1]: Cambiar a minuscula

Comentado [LG2R1]: Corregido

SOFTWARE:

- Alojamiento del servidor.
- Hosting.
- Servidor web apache.
- Sistema operativo mínimo Windows 7 hasta la versión actual Windows 10 “20.04”.
- Aplicación web adaptable.

Comentado [MOU3]: Cambiar a minúscula

1.3.Requerimientos de la aplicación.

Para el desarrollo del presente software de simulación del Examen Complexivo para evaluar los conocimientos de los estudiantes, se definieron los siguientes puntos de forma en general para que los docentes y estudiantes puedan cubrir las necesidades que requiere para el desarrollo de la presente propuesta. En donde los estudiantes puedan sentirse seguro de poder rendir el Examen Complexivo.

1.3.1. Definición de actores.

Un actor es la persona que interactuará con **SOFTWARE DE SIMULACION DEL EXAMEN COMPLEXIVO PARA EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTES DE LA FACULTAD DE AMINISTRACION FINANZAS E INFORMATICA DE LA UTB**, de acuerdo a ciertas características y permisos estipulados por el administrador del software.

Por lo anterior se reconocen los siguientes actores:

1.3.2. Administrador

Es la persona o usuario que tendrá las posibilidades de dar administración al sistema, dentro de esta administración se nombra características de:

- Ingreso, consulta y edición de administradores;
- Ingreso, consulta y edición de docentes;
- Ingreso, consulta y edición de estudiantes;
- Ingreso, consulta y edición de áreas;
- Ingreso, consulta y edición de sub-áreas;

Activar o desactivar administradores, docentes, estudiantes, áreas o sub-áreas, igualmente podrá consultar los exámenes, preguntas y resultados de los exámenes presentados por los estudiantes.

1.3.3. Docente

Este actor se caracteriza por ser aquel al que se le permite ingresar, consultar o editar exámenes; ingresar, consultar o editar preguntas de los exámenes existentes en el sistema; también podrá activar o desactivar exámenes para ser presentados por los estudiantes; podrá consultar estudiantes, áreas, sub-áreas y resultados de exámenes presentados.

1.3.4. Estudiante

Este rol únicamente podrá consultar áreas, exámenes y resultados de los mismos una vez se haya inscrito para la presentación de uno o varios exámenes.

1.4. Justificación del requerimiento a satisfacer.

Ante la necesidad de aprendizaje diario, y teniendo en cuenta la importancia de la formación académica de estudiantes que están por culminar su carrera, el proyecto se hace sustancial para ayudar al estudiante a encontrar sus debilidades y/o fortalezas en ciertos temas, igualmente la información generada dará pautas tanto para docentes como para personal administrativo en busca de plantear mejoras en los procesos de conceptualización de la información y el manejo que le es dado.

Por tanto, se plantea el presente **SOFTWARE DE SIMULACIÓN DEL EXAMEN COMPLEXIVO PARA EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTES DE LA FACULTAD DE ADMINISTRACIÓN FINANZAS E INFORMÁTICA DE LA UTB** con el fin de dar a los usuarios mejor manejo de la información, seguridad en la información, interacción con un sistema amigable y sin demasiadas restricciones como lo hacen sistemas que cumplen la misma función.

CAPITULO II

2. Desarrollo del Prototipo.

Para realizar el presente software se analizaron los requerimientos que se expusieron en el capítulo anterior.

2.1. Definición del Prototipo Tecnológico.

Desarrollo del prototipo del software de simulación del examen complejo para evaluar los conocimientos de los estudiantes de la facultad de administración finanzas e informática de la UTB.

2.2. Fundamentación Teórica del Prototipo.

La aplicación web permitirá tanto a los estudiantes puedan actualizar ciertos campos para poder rendir el examen complejo, y por lo tanto el administrador y los docentes puedan gestionar la actualización de la base de datos de las preguntas a tomar.

2.2.1. Sistema informático Web.

Los sistemas informáticos también llamados (SI), son utilizados para guardar y almacenar información y están compuesto por hardware, software y los seres humanos.

Los sistemas informáticos web son enfocados en la red, los usuarios pueden acceder a través de internet, con la ayuda de los distintos servidores que componen el sistema informático web, a ellos se accede por medio de los navegadores web.

2.2.2. Alojamiento de Registro.

Tipos de alojamiento de los registros: hay varios tipos de distintos de alojamiento de los registros que también hay que tener en cuenta y son importantes cuando se quiere seleccionar un

sistema de almacenamiento.(*Gestión de Servicios En El Sistema Informático. IFCT0509 - Ester Chicano Tejada - Google Libros, n.d.*)

Alojamiento tradicional: el alojamiento tradicional de datos se utiliza cuando la organización dispone en sus instalaciones de equipos destinados al almacenamiento. La organización auto gestiona al almacenamiento de sus registros.(*Gestión de Servicios En El Sistema Informático. IFCT0509 - Ester Chicano Tejada - Google Libros, n.d.*)

Alojamiento web o “web hosting”: el alojamiento web es un tipo de almacenamiento en los que los datos y registro se encuentra almacenado en internet (páginas web, servidores, etc.) y se puede acceder a ellos de modo virtual desde cualquier equipo o dispositivos. Ese tipo de alojamiento puede ser gratuito, aunque no es habitual y los servicios ofrecidos están bastante limitados. Suelen ser alojamientos de pagos en los que se alquila espacio de almacenamiento en un disco virtual o en sitio web.(*Gestión de Servicios En El Sistema Informático. IFCT0509 - Ester Chicano Tejada - Google Libros, n.d.*)

2.2.3. Gestión de Datos.

Definición de los datos, seguridad e integridad de datos, manipulación de los datos y concurrencia de los datos.

Es la agrupación de recursos y personas que se encargan de procesar la información. Implica la recolección, acumulación, ejecución y recuperación de información.(*..3 Objetivos Específicos de La Gestión de Datos, 1999*)

2.2.4. Requerimientos.

Un requerimiento determina lo que el sistema debe de hacer, que funciones tiene que realizar y propiedades necesarias y deseables. La captura de los requerimientos tiene como principal objetivo llevar a un entendimiento profundo y no debe hacer el sistema que se está desarrollando. La validación de requerimientos permite comprobar que estos definen el sistema que el cliente busca, asegurando que son completos, exactos consistentes. (*Desarrollo y Optimización de Componentes Software Para Tareas ... - José Luis Villada Romero - Google Libros, n.d.*)

2.2.5. Requerimientos Operacionales, Funcionales y de Rendimiento por:

2.2.5.1. Ingreso Primario Administrador.

A este ingreso solo puede acceder el Docente autorizada estipulado por la **FACULTAD ADMINISTRATIVA, FINANZAS E INFORMATICA DE LA UTB**. Debido que podrá gestionar lo siguiente Ítems (insertar, eliminar, buscar, modificar, etc.) todo lo que tenga que ver **con las peticiones de los usuarios, para la ejecución del Sistema web para la toma de simulación** del Examen Complexivo, para poder ingresar deberá escribir el login del sistema la palabra **ADMIN** y luego la clave correspondiente.

2.2.5.2. Ingreso Secundario. (Docente)

En este permitirá ingresar a cualquier docente que se asigne para la simulación de la toma del Examen Complexivo, para poder ingresar deberá escribir el login del sistema el usuario la clave que se le ha sido asignada por el sistema, y si es un usuario nuevo que por primera vez deberá registrarse.

2.2.5.3. Ingreso Secundario. (Estudiante)

En este ingreso permitirá que el estudiante pueda realizar la simulación de la toma del Examen Complexivo, para poder ingresar el estudiante deberá escribir en el login la palabra estudiante y la clave que haya sido asignada una vez que me haya registrado previamente o que el administrador del sistema le haya creado el usuario y la contraseña.

2.2.5.4. Ingreso de Datos.

2.2.5.4.1. Ingreso de Datos del Docente y Estudiantes.

Para poder ingresar datos de los docentes y estudiantes el administrador debe de ingresar a las pestañas crear usuario y poner toda la información y también deberá asignarle los permisos que según el usuario debe tener.

2.2.5.5. Modificar Datos.

El administrador realiza la modificación de datos ya sea en los docentes asignados y también en los estudiantes. Se debe escoger en editar y hacer los cambios que sean necesarios a cambiar.

2.2.5.6. Eliminar Datos.

El administrador puede eliminar datos ya sean de los docentes asignados y estudiantes. Se debe escoger la opción eliminar. Lo que sea necesario en eliminar.

2.2.5.7. Búsqueda de Datos.

El administrador puede buscar datos tanto como los docentes asignados y estudiantes. Se utiliza la opción búsqueda.

2.2.6. Diagrama de Caso de Uso.

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.

Los diagramas de caso de uso modelan la funcionabilidad del sistema usando actores y casos de uso. Los casos de uso son de servicios o funciones provista por el sistema para sus usuarios.(Armando Canchala, n.d.)

2.2.6.1.Símbolos De Los Casos de Uso.

Sistema: el rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican afuera de los límites del sistema.(Armando Canchala, n.d.)

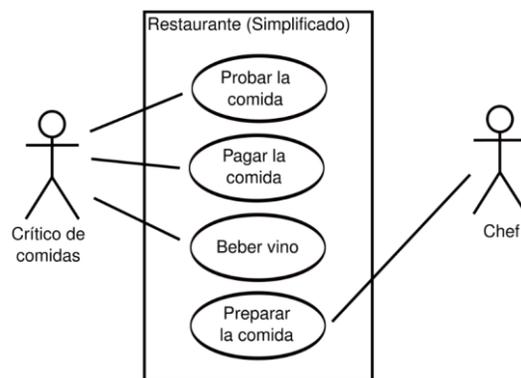


Figura 2: Sistema de Casos de Uso
Fuente.(Armando Canchala, n.d.):

Actor: Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.(Armando Canchala, n.d.)

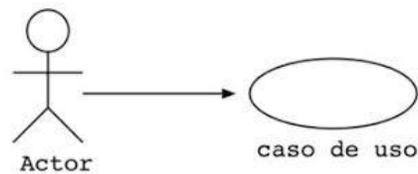


Figura 4: Sistema de Casos de Uso
Fuente.(Armando Canchala, n.d.)

Por ejemplo:

Un actor puede ser empleado, pero también puede ser cliente en la de la empresa. Incluso cuando la misma persona en el mundo real, se representa como dos símbolos distintos en un diagrama de caso de uso, ya que la persona interactúa con el sistema en distintos roles.

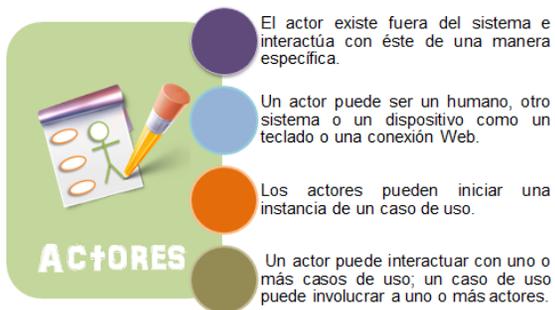


Figura 5: Características de los actores
Fuente.(Armando Canchala, n.d.)

2.2.6.2.Relaciones.

Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “Incluir” o “extender”. Una relación “incluir” indica que es un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “extender” indica opciones alternativas para un cierto caso de uso.

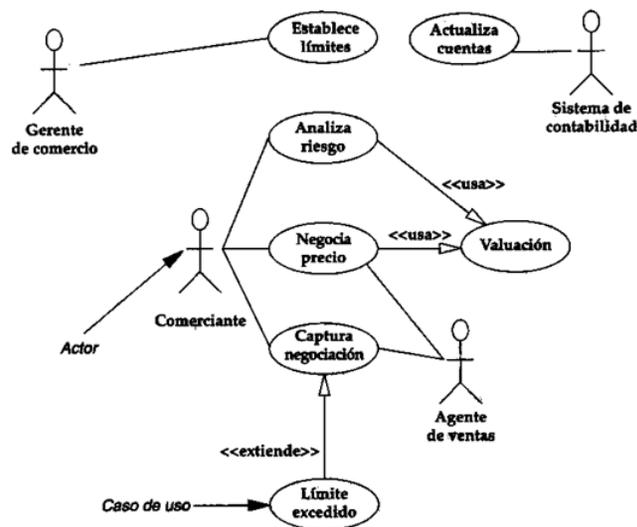


Figura 6: Ejemplo de caso de uso
Fuente.(Armando Canchala, n.d.)

2.2.6.3.Relaciones de los Casos de Uso.

Las relaciones activas se conocen como relaciones de comportamiento y se utilizan principalmente en los diagramas de casos de uso. Hay cuatro tipos básicos de relaciones de comportamiento: comunica, incluye, extiende y generaliza.

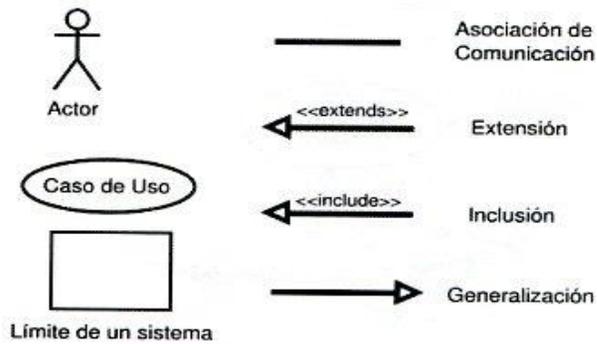


Figura 7: Ejemplo de caso de uso
Fuente.(Armando Canchala, n.d.)

2.2.6.4. Documentación de los Casos de Uso.

Existen dos formas principales de documentar un caso de uso:

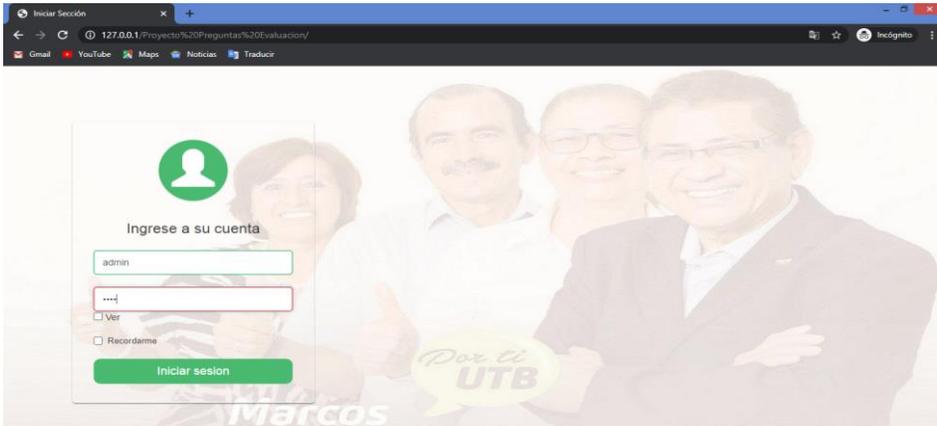
- ✓ Un diagrama en UML.
- ✓ Un documento detallado.

Documentar casos de usos no es una tarea fácil que se pueda dominar de un día para otro, requiere de tiempo, disciplina y experiencia, sin embargo, podemos definir una serie de pasos identificables para escribir los casos de uso.

2.2.7. Definición de Casos de Uso de los requerimientos.

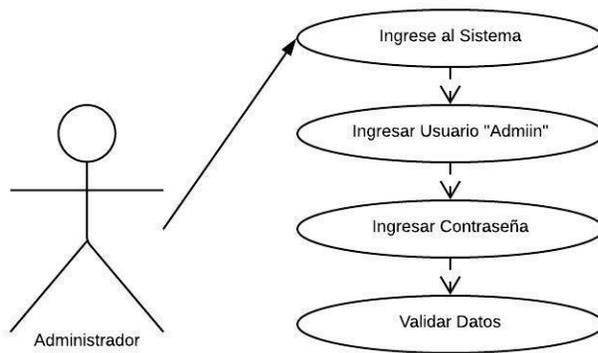
Los casos de usos es una secuencia de transacciones que son desarrollados por unos sistemas en respuesta a un evento que inicia un actor sobre el propio sistema.

2.2.7.1. Ingreso del Sistema Informático Web.



2.2.7.1.1. Ingreso primario (Administrador).

Figura 8: Ingreso Primario Administrador



Fuente: Elaborado por el autor.

2.2.7.1.2. Ingreso Secundario (Docente).

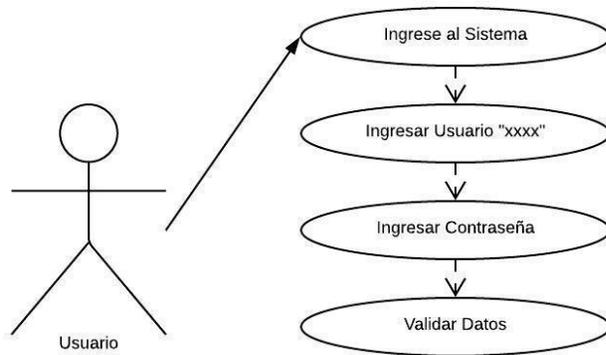


Figura 9: Ingreso Primario Administrador
Fuente: Elaborado por el autor.

2.2.7.1.3. Ingreso Secundario (Estudiantes).

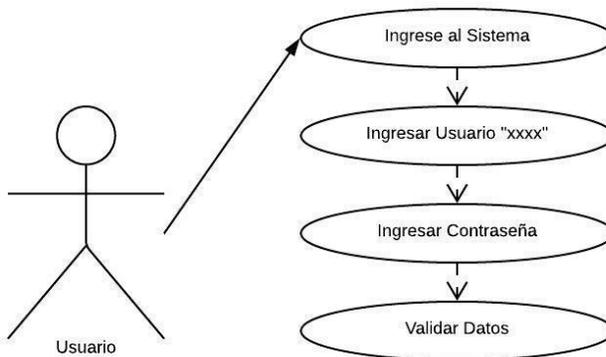


Figura 10: Ingreso Secundario estudiantes
Fuente: Elaborado por el autor.

2.2.7.2.Ingreso de Datos del Usuario.

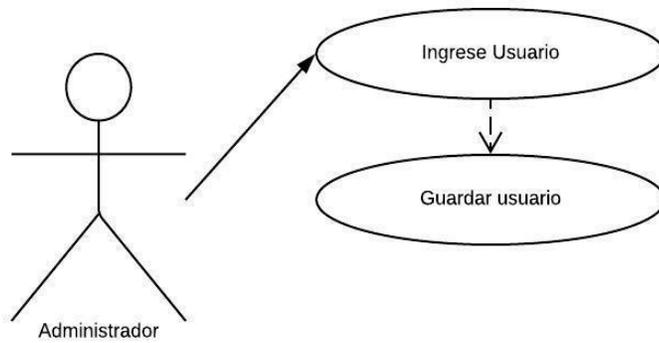


Figura 11: Ingreso de los usuarios
Fuente: Elaborado por el autor.

2.2.7.3.Modificar Datos.

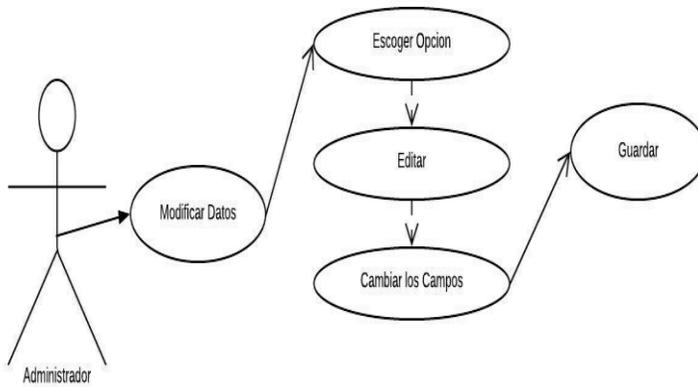


Figura 12: Modificación de datos
Fuente: Elaborado por el autor.

2.2.7.4. Eliminar Datos.

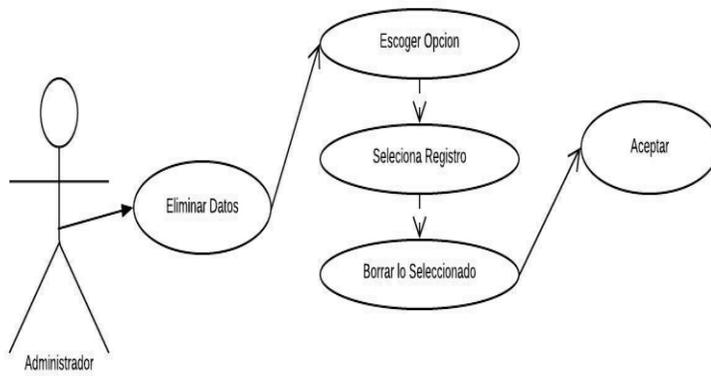


Figura 13: Eliminar datos
Fuente: Elaborado por el autor.

2.2.7.5. Búsqueda de Datos.

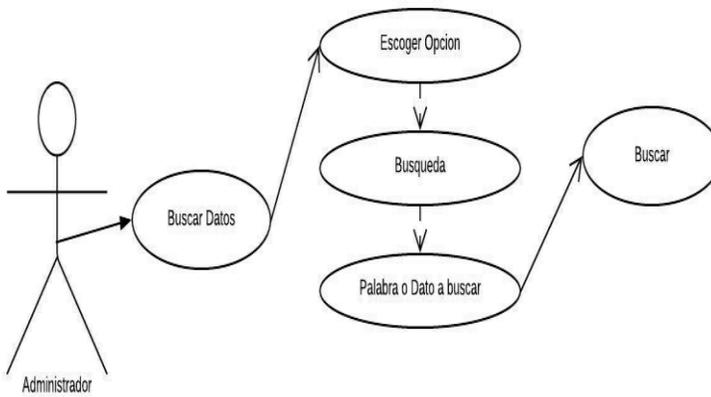


Figura 14: Búsqueda de datos
Fuente: Elaborado por el autor.

2.2.8. Metodología RAD.

El desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de rapid application development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1991 basado en el trabajo hecho por Scott Shutz en los 80. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades Case (Ingeniería asistida por computadora). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad y la rapidez de ejecución.(Martin & Martin, 1990).

2.2.8.1.Fases de la Metodologías RAD.

Comenzar con el desarrollo rápido de aplicaciones generalmente sigue un proceso cíclico que incluye cuatros pasos básicos:(James Martin, n.d.)

Planificación de Requerimientos: durante esta etapa inicial diseñadores, desarrolladores y usuarios llegan a un acuerdo aproximado sobre el alcance de los proyectos y los requisitos de la aplicación, para que puedan comenzar las etapas con creación prototipos.(James Martin, n.d.)

Diseño con el usuario: Los comentarios de los usuarios se recopilan con gran énfasis en la determinación de la arquitectura del sistema. Estos permiten crear modelos y prototipos iniciales. Este paso se repite tantas veces como sea necesario a medida que el proyecto evoluciona.(James Martin, n.d.)

Construcción: Una vez que ha comenzado el diseño básico del usuario y del sistema, la fase de construcción es donde se lleva a cabo la mayor parte de la codificación, las pruebas y la integración reales de la aplicación. Junto con el diseño del usuario la fase de construcción rápida

se repite tantas veces como sea necesario, a medida que se requieran nuevos componentes o se realicen modificaciones para satisfacer las necesidades del proyecto. (James Martin, n.d.)

Transición: La etapa final de transición (o Cutover) le permite al equipo de desarrollo tiempo para mover los componentes a un entorno de producción en vivo, donde se pueden llevar a cabo todas las pruebas necesarias o la capacitación del equipo.(James Martin, n.d.)



2.2.9. UML (Lenguaje Unificado de Modelado)

El lenguaje unificado de modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas.(James Rumbaugh, Ivar Jacobson, 2000)

Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre las técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación y principios generales.(James Rumbaugh, Ivar Jacobson, 2000)

Tiene partes estáticas, dinámicas de entornos y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo interactivo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.(James Rumbaugh, Ivar Jacobson, 2000)

2.2.9.1.¿Qué es y para qué sirve UML?

El término “lenguaje” ha generado bastante confusión respecto a lo que es un UML. En realidad, el termino lenguaje quizás no es el más apropiado, ya que no es un lenguaje propiamente dicho, sino una serie de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software. Mucha gente piensa por confusión que el UML es un lenguaje de programación y esta idea es errónea: (Cesar Krall, 2006)

UML no es un lenguaje de programación, como decimos, UML son una serie de normas y estándares que dicen cómo se debe representar algo.(Cesar Krall, 2006)

UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y frecuentemente usada por analista funcionales (aquellos que define que debe hacer un programa sin entra a escribir código) y analistas-programadores (aquellos dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python, o cualquier otro). Por tanto, si estás dando tus primeros pasos en programación, te recomendaríamos que te olvides de UML hasta que tengas los conocimientos mínimos como se usa de condicionales, bucles y conocimientos de la programación orientadas a objetos. Esto es solo una recomendación, realidad prácticamente cualquier persona puede usar el UML, incluso podría

usarse para realizar esquemas o documentación de procesos que no tenga que ver con la informática. (Cesar Krall, 2006)

2.2.10. Método de Análisis Estructurado

Muchos especialistas en sistemas e información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo de análisis estructurado tiene como finalidad superar esta dificultad por medio de:

La división del sistema en sus componentes o la construcción de un modelo del sistema. Este método se incorpora elementos tanto de análisis como diseño.(Jhon Neva, 2012)

2.2.10.1. ¿Qué es el Análisis Estructurado?

El análisis estructurado se encuentra en especificar lo que se requiere del sistema. No se establece como se cumplirán los requerimientos ni la forma en que se implantara la aplicación, más bien permite que las personas estudien los elementos lógicos (lo que hace el sistema). Separado de los componentes físicos (PC, terminales, sistemas de almacenamientos).(Jhon Neva, 2012)

2.2.11. PHP

La sigla PHP identifica a un lenguaje de programación que nació como personal home page (PHP) tolos. Fue desarrollado por el programador de origen danés **Rasmus Lerdorf** en 1994 con el propósito de facilitar el diseño de páginas web de carácter dinámico. (Julián Perez Porto, 2010)

El acrónimo recursivo, sin embargo, en la actualidad está vinculado a PHP Hypertext Pre-Processor. El lenguaje es desarrollado hoy en día por The PHP Group aunque carece de una normativa formal. La Free Software Foundation

por lo tanto, considera la licencia PHP como parte del software libre.(Julián Perez Porto, 2010)

2.2.11.1. Lenguaje de Programación de PHP

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del servidor que se podían incorporar directamente en el documento HTML en el lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usado en aplicaciones graficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo(Wictea, 2015).

2.2.11.2. Características de PHP

- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.(Wictea, 2015)
- ✓ Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de PHP Arrays.(Wictea, 2015)
- ✓ El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.(Wictea, 2015)
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.(Wictea, 2015)
- ✓ Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).(Wictea, 2015)
- ✓ Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están aplicadas y ejemplificadas en un único archivo de ayuda.(Wictea, 2015)
- ✓ El libre, por lo que se presenta como una alternativa de fácil acceso para todos. (Wictea, 2015)
- ✓ Permite aplicar técnicas de programación orientada a objetos.(Wictea, 2015)
- ✓ No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.(Wictea, 2015)

- ✓ Tiene manejo de excepciones (desde PHP5).(Wictea, 2015)
- ✓ Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones web de manejo de contenido, y es su uso principal.(Wictea, 2015)

2.2.12. Servidor Wamp.

Servidor Wamp es un acrónimo que significa Windows, apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que significa que cuando instala WAMP, estas instalando en tu sistema operativo (Windows en caso de WAMP). Aunque puede instalarlos por separado, por lo general son empaquetados, y también por una buena razón.(Gustavo B, 2019)

Lo que es bueno saber es que WAMP deriva de LAMP (la L significa Linux). La única diferencia entre estos dos es que WAMP se usa para Windows, mientras que LAMP para sistemas operativos basados en Linux.(Gustavo B, 2019)

2.2.12.1. NotePad++

NotePad++ es un editor de código fuente gratuito (como en “discurso libre” y también como en “cerveza gratis”) y reemplazo del bloc de notas que admite varios idiomas. Ejecutando en el entorno MS Windows, su uso rige por la licencia GPL.

Basado en el poderoso componente de Scintilla, NotePad++ está escrito en C ++ y usa API win32 pura y STL que asegura una mayor velocidad de ejecución y un tamaño de programa más pequeño. Al optimizar tantas rutinas como sea posible sin perder la facilidad de uso, NotePad ++ está tratando de reducir las emisiones mundiales de dióxido de carbono. Cuando se utiliza menos

energía de la CPU, la PC puede reducir la velocidad y reducir el consumo de energía, lo que da como resultado un entorno más ecológico.(Jimena, 2018)

2.2.12.2. Características

- ✓ Resaltos de sintaxis y envoltura de sintaxis.
- ✓ Resaltado de sintaxis definido por el usuario.
- ✓ Auto-completado.
- ✓ Multi-documento.
- ✓ Multi-visor.
- ✓ Soporte para la búsqueda y sustitución de expresiones regulares.
- ✓ Soporte completo para arrastrar y soltar.
- ✓ Posición dinámica de vistas.
- ✓ Auto- detección de estado del fichero.
- ✓ Aumentar y reducir el zoom.
- ✓ Entorno Multi-lenguaje soportado
- ✓ Favoritos.
- ✓ Resaltado de paréntesis y sangría.
- ✓ Grabación y ejecución de macros.

2.2.13. PostgreSQL



2.2.13.1. ¿Qué es PostgreSQL?

PostgreSQL es una de las opciones más interesantes en bases de datos relacionales Open-Source. Michael Stonebraker inició el proyecto bajo el nombre Post Ingres a mediados de los 80's con la idea de solucionar problemas existentes en las bases de datos en esa época. MySQL fue por mucho tiempo el motor más popular; pero hoy es propiedad de Oracle y esto limita su evolución.(Reicek, 2014)

Es gratuito y libre, además de que hoy nos ofrece una gran cantidad de opciones avanzadas. De hecho, es considerado el motor de base de datos más avanzado en la actualidad.(Reicek, 2014)

Una característica interesante de PostgreSQL es el control de concurrencias multiversión; o MVCC por sus siglas en inglés. Este método agrega una imagen del estado de la base de datos a cada transacción. Esto nos permite hacer transacciones eventualmente consistentes, ofreciéndonos grandes ventajas en el rendimiento.(Reicek, 2014)

En PostgreSQL no se requiere usar bloqueos de lectura al realizar una transacción lo que nos brinda una mayor escalabilidad. También PostgreSQL tiene Hot-Standby. Este permite que los clientes hagan búsquedas (sólo de lectura) en los servidores mientras están en modo de recuperación o espera. Así podemos hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema.(Reicek, 2014)

PostgreSQL aporta mucha flexibilidad a nuestros proyectos. Por ejemplo, nos permite definir funciones personalizadas por medio de varios lenguajes. Algunos son:(Reicek, 2014)

- ✓ PL/pgSQL
- ✓ PL/Tcl
- ✓ PL/Perl
- ✓ PL/Python
- ✓ PL/PHP
- ✓ PL/Ruby
- ✓ PL/Java

2.2.14. WampServer.



Wamp es un acrónimo que significa Windows, apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que significa que cuando instalas estás instalando Apache, MySQL y PHP en tu sistema operativo (Windows en el caso de WAMP). Aunque puede instalarlos por separado, por lo general son empaquetados, y también por una buena razón.

Lo que es bueno saber es que WAMP deriva de LAMP (la L significa Linux). La única diferencia entre estos dos es que WAMP se usa para Windows, mientras que LAMP para sistemas operativos basados en Linux.

Repasemos rápidamente lo que representa cada letra:

«W» significa Windows; también hay LAMP (para Linux) y MAMP (para Mac).

«A» significa Apache. Apache es el software de servidor que se encarga de servir las páginas web. Cuando solicitas ver una página, Apache cumple tu solicitud a través de HTTP y te muestra el sitio.

«M» significa MySQL. El trabajo de MySQL es ser el sistema de gestión de base de datos para tu servidor. Almacena toda la información relevante, como el contenido de tu sitio, los perfiles de usuario, etc.

«P» significa PHP. Es el lenguaje de programación en el cual está escrito WordPress y actúa como aglutinante para toda este stack de soluciones. PHP se ejecuta junto con Apache y se comunica con MySQL.

2.3.Objetivos del Prototipo.

2.3.1. Objetivo General.

Desarrollar un sistema web informático para EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTE DE LA FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICA DE LA UTB.

2.3.2. Objetivo Específico.

- ✓ Obtener los datos necesarios mediante las técnicas de recopilación de información aplicadas en el caso de estudio.
- ✓ Realizar el modelo entidad relación de la base de datos del sistema de evaluación.
- ✓ Desarrollo del diagrama físico de todos los procesos que se realizan en el sistema de evaluación.
- ✓ Diseñar y codificar el sistema web para EVALUAR LOS CONOCIMIENTOS DE LOS ESTUDIANTE DE LA FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICA DE LA UTB.

2.4. Diseño del Prototipo.

2.4.1. Diagrama Entidad Relación.

Ingreso al Sistema Informático

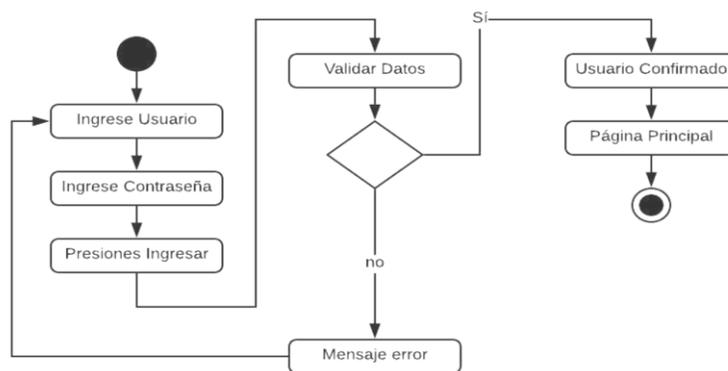


Figura 22: Ingreso al SI.
Fuente: Elaborado por el autor.

Ingreso del Administrador de un nuevo Usuario

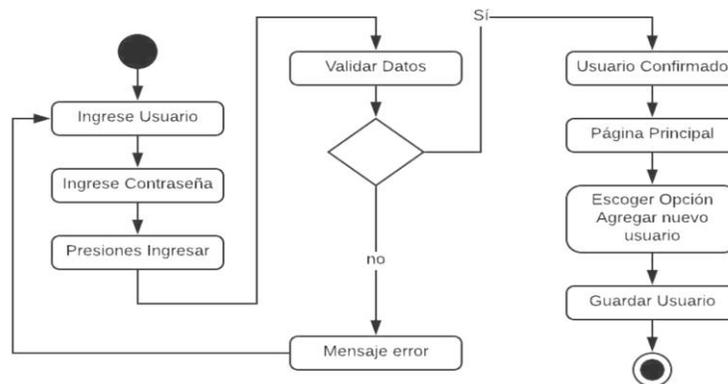


Figura 23: Ingreso al SI como administrador.
Fuente: Elaborado por el autor.

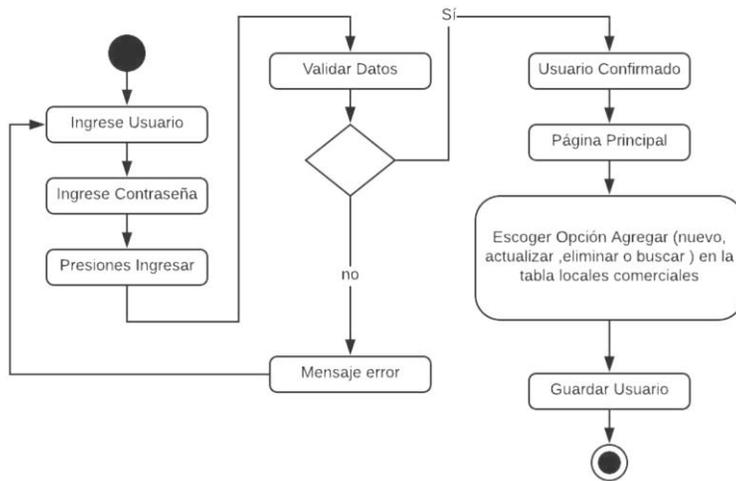
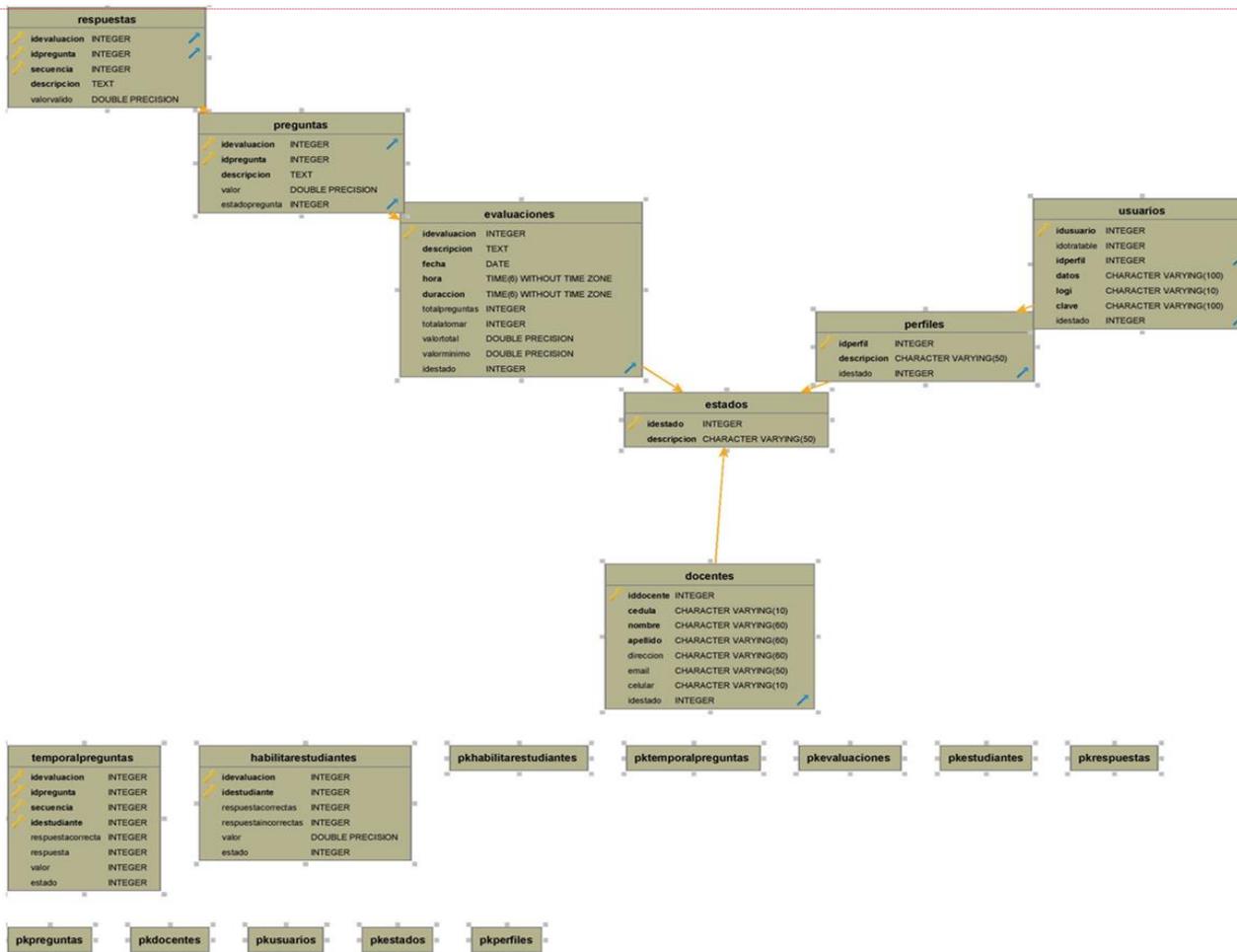


Figura 24: Ingreso al SI como usuario.
Fuente: Elaborado por el autor.

2.4.2. Modelo de Entidad Relación.

Comentado [MOU4]: debe ir MER modeleo entidad relación



2.4.3. Script de la base de datos.

```
-- Table: docentes
-- DROP TABLE docentes;
CREATE TABLE docentes
(
  iddocente integer NOT NULL,
  cedula character varying(10) NOT NULL,
  nombre character varying(60) NOT NULL,
  apellido character varying(60) NOT NULL,
  direccion character varying(60),
  email character varying(50),
  celular character varying(10),
  idestado integer DEFAULT 1,
  CONSTRAINT pkdocentes PRIMARY KEY (iddocente),
  CONSTRAINT usuariosfkestados FOREIGN KEY (idestado)
  REFERENCES estados (idestado) MATCH SIMPLE
  ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
  OIDS=FALSE
);
ALTER TABLE docentes
  OWNER TO postgres;

-- Trigger: deletedocente on docentes
-- DROP TRIGGER deletedocente ON docentes;

CREATE TRIGGER deletedocente
  AFTER DELETE
  ON docentes
  FOR EACH ROW
  EXECUTE PROCEDURE deleteusuariodocente();

-- Trigger: insertdocente on docentes

-- DROP TRIGGER insertdocente ON docentes;
CREATE TRIGGER insertdocente
  AFTER INSERT
  ON docentes
  FOR EACH ROW
  EXECUTE PROCEDURE insertusuariodocente();

-- Trigger: updatedocente on docentes
-- DROP TRIGGER updatedocente ON docentes;
```

```
CREATE TRIGGER updatedocente
  AFTER UPDATE
  ON docentes
  FOR EACH ROW
  EXECUTE PROCEDURE updateusuariodocente();

-- Table: estados
-- DROP TABLE estados;

CREATE TABLE estados
(
  idestado integer NOT NULL,
  descripcion character varying(50) NOT NULL,
  CONSTRAINT pkestates PRIMARY KEY (idestado)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE estados
  OWNER TO postgres;

-- Table: estudiantes
-- DROP TABLE estudiantes;

CREATE TABLE estudiantes
(
  idestudiante integer NOT NULL,
  cedula character varying(10) NOT NULL,
  nombre character varying(60) NOT NULL,
  apellido character varying(60) NOT NULL,
  direccion character varying(60),
  email character varying(50),
  celular character varying(10),
  idestado integer DEFAULT 1,
  CONSTRAINT pkeestudiantes PRIMARY KEY (idestudiante),
  CONSTRAINT usuariosfkestates FOREIGN KEY (idestado)
    REFERENCES estados (idestado) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
  OIDS=FALSE
);
ALTER TABLE estudiantes
  OWNER TO postgres;
```

```

-- Trigger: deleteestudiante on estudiantes
-- DROP TRIGGER deleteestudiante ON estudiantes;

CREATE TRIGGER deleteestudiante
AFTER DELETE
ON estudiantes
FOR EACH ROW
EXECUTE PROCEDURE deleteusuarioestudiante();

-- Trigger: insertestudiante on estudiantes
-- DROP TRIGGER insertestudiante ON estudiantes;

CREATE TRIGGER insertestudiante
AFTER INSERT
ON estudiantes
FOR EACH ROW
EXECUTE PROCEDURE insertusuarioestudiante();

-- Trigger: updateestudiante on estudiantes
-- DROP TRIGGER updateestudiante ON estudiantes;

CREATE TRIGGER updateestudiante
AFTER UPDATE
ON estudiantes
FOR EACH ROW
EXECUTE PROCEDURE updateusuarioestudiante();

-- Table: evaluaciones
-- DROP TABLE evaluaciones;

CREATE TABLE evaluaciones
(
  idevaluacion integer NOT NULL,
  descripcion text NOT NULL,
  fecha date NOT NULL,
  hora time without time zone NOT NULL,
  duracion time without time zone NOT NULL,
  totalpreguntas integer DEFAULT 0,
  totalatimar integer DEFAULT 0,
  valortotal double precision DEFAULT 0,
  valorminimo double precision DEFAULT 0,
  idestado integer DEFAULT 1,
  CONSTRAINT pkevaluaciones PRIMARY KEY (idevaluacion),
  CONSTRAINT evaluacionfkestados FOREIGN KEY (idestado)

```

```

REFERENCES estados (idestado) MATCH SIMPLE
ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
  OIDS=FALSE
);
ALTER TABLE evaluaciones
  OWNER TO postgres;

-- Trigger: calcularpreguntasupdate on evaluaciones

-- DROP TRIGGER calcularpreguntasupdate ON evaluaciones;

CREATE TRIGGER calcularpreguntasupdate
  AFTER UPDATE
  ON evaluaciones
  FOR EACH ROW
  EXECUTE PROCEDURE calcularpreguntastomarupdate();

-- Table: habilitarestudiantes
-- DROP TABLE habilitarestudiantes;

CREATE TABLE habilitarestudiantes
(
  idevaluacion integer NOT NULL,
  idestudiante integer NOT NULL,
  respuestacorrectas integer DEFAULT 0,
  respuestaincorrectas integer DEFAULT 0,
  valor double precision DEFAULT 0,
  estado integer DEFAULT 4,
  CONSTRAINT pkhabilitarestudiantes PRIMARY KEY (idevaluacion, idestudiante)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE habilitarestudiantes
  OWNER TO postgres;

-- Table: perfiles
-- DROP TABLE perfiles;

CREATE TABLE perfiles
(
  idperfil integer NOT NULL,
  descripcion character varying(50) NOT NULL,

```

```

idestado integer DEFAULT 1,
CONSTRAINT pkperfiles PRIMARY KEY (idperfil),
CONSTRAINT perfilesfkestados FOREIGN KEY (idestado)
    REFERENCES estados (idestado) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
    OIDS=FALSE
);
ALTER TABLE perfiles
    OWNER TO postgres;

-- Table: preguntas
-- DROP TABLE preguntas;

CREATE TABLE preguntas
(
    idevaluacion integer NOT NULL,
    idpregunta integer NOT NULL,
    descripcion text NOT NULL,
    valor double precision DEFAULT 0,
    estadopregunta integer DEFAULT 1,
    CONSTRAINT pkpreguntas PRIMARY KEY (idevaluacion, idpregunta),
    CONSTRAINT evaluacionesfkpreguntas FOREIGN KEY (idevaluacion)
        REFERENCES evaluaciones (idevaluacion) MATCH SIMPLE
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT evaluacionfkestados FOREIGN KEY (estadopregunta)
        REFERENCES estados (idestado) MATCH SIMPLE
        ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
    OIDS=FALSE
);
ALTER TABLE preguntas
    OWNER TO postgres;

-- Trigger: calcularpreguntasdelete on preguntas
-- DROP TRIGGER calcularpreguntasdelete ON preguntas;

CREATE TRIGGER calcularpreguntasdelete
    AFTER DELETE
    ON preguntas
    FOR EACH ROW

```

```

EXECUTE PROCEDURE calcularpreguntastomardelete();

-- Trigger: calcularpreguntasinsert on preguntas
-- DROP TRIGGER calcularpreguntasinsert ON preguntas;

CREATE TRIGGER calcularpreguntasinsert
AFTER INSERT
ON preguntas
FOR EACH ROW
EXECUTE PROCEDURE calcularpreguntastomarinsert();

-- Table: respuestas
-- DROP TABLE respuestas;

CREATE TABLE respuestas
(
  idevaluacion integer NOT NULL,
  idpregunta integer NOT NULL,
  secuencia integer NOT NULL,
  descripcion text NOT NULL,
  valorvalido double precision DEFAULT 0,
  CONSTRAINT pkrespuestas PRIMARY KEY (idevaluacion, idpregunta, secuencia),
  CONSTRAINT preguntasfkrespuestas FOREIGN KEY (idevaluacion, idpregunta)
    REFERENCES preguntas (idevaluacion, idpregunta) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
  OIDS=FALSE
);
ALTER TABLE respuestas
OWNER TO postgres;

- Table: temporalpreguntas

-- DROP TABLE temporalpreguntas;

CREATE TABLE temporalpreguntas
(
  idevaluacion integer NOT NULL,
  idpregunta integer NOT NULL,
  secuencia integer NOT NULL,
  idestudiante integer NOT NULL,
  respuestacorrecta integer DEFAULT 0,

```

```
    respuesta integer DEFAULT 0,
    valor integer DEFAULT 0,
    estado integer DEFAULT 4,
    CONSTRAINT pktemporalpreguntas PRIMARY KEY (idevaluacion, idpregunta,
    secuencia, idestudiante)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE temporalpreguntas
    OWNER TO postgres;
```

```
-- Table: usuarios
-- DROP TABLE usuarios;
```

```
CREATE TABLE usuarios
(
    idusuario integer NOT NULL,
    idotratable integer DEFAULT 0,
    idperfil integer NOT NULL,
    datos character varying(100) NOT NULL,
    logi character varying(10) NOT NULL,
    clave character varying(100) NOT NULL,
    idestado integer DEFAULT 1,
    CONSTRAINT pkusuarios PRIMARY KEY (idusuario),
    CONSTRAINT usuariosfk estados FOREIGN KEY (idestado)
        REFERENCES estados (idestado) MATCH SIMPLE
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT usuariosfkperfiles FOREIGN KEY (idperfil)
        REFERENCES perfiles (idperfil) MATCH SIMPLE
        ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (
    OIDS=FALSE
);
ALTER TABLE usuarios
    OWNER TO postgres;
```

2.5.Ejecución y/o Ensamblaje del Prototipo.

2.5.1. Plan de Instalación.

La instalación del sistema informático cuando lo tenemos de forma local no necesitamos algún sistema operativo específico solo basta tener instalado el servidor Wamp y si lo hemos desarrollado en otro ordenador se debe pegar el proyecto en la carpeta C:\wamp\www.

Instalación del Wamp server primero tienes que descargarlo de la página oficial en español <http://www.wampserver.es/>.

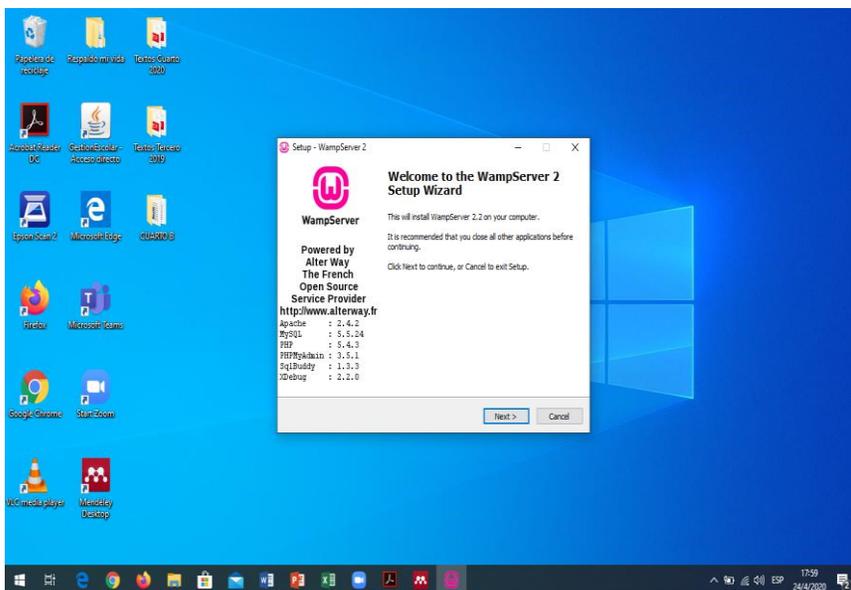


Figura: Instalación de Wamp server en el Sistema Operativo.

Fuente: Elaborado por el autor.

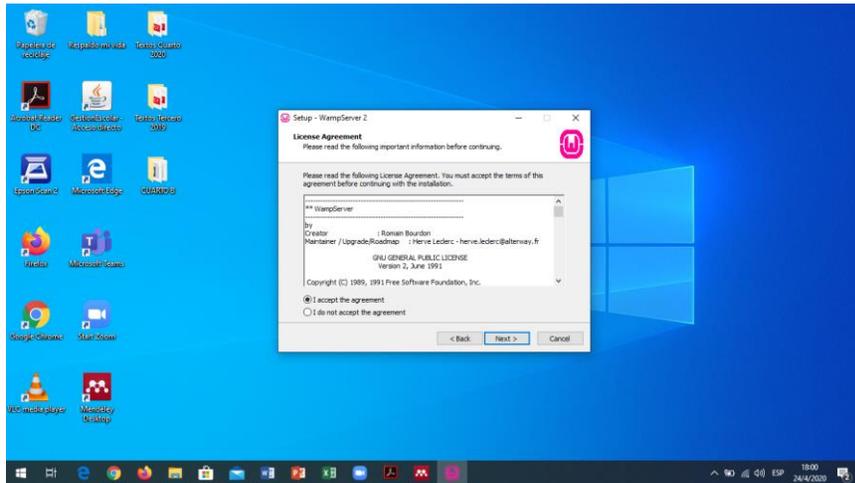


Figura: Aceptamos los términos.
Fuente: Elaborado por el autor.

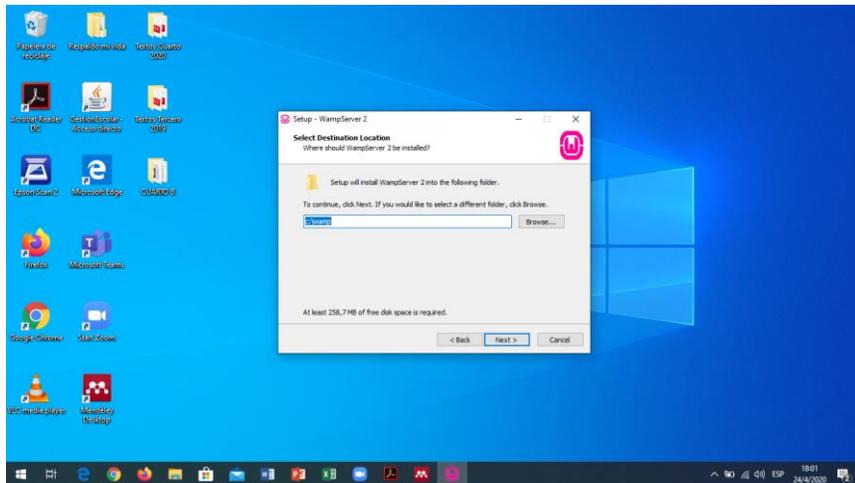


Figura: Selección de la carpeta Wamp.
Fuente: Elaborado por el autor.

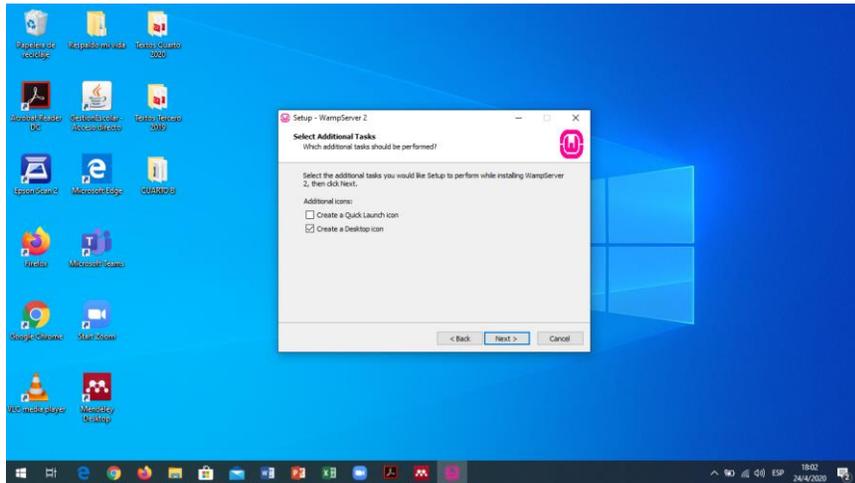


Figura: Selección de poner el icono en el escritorio.
Fuente: Elaborado por el autor.

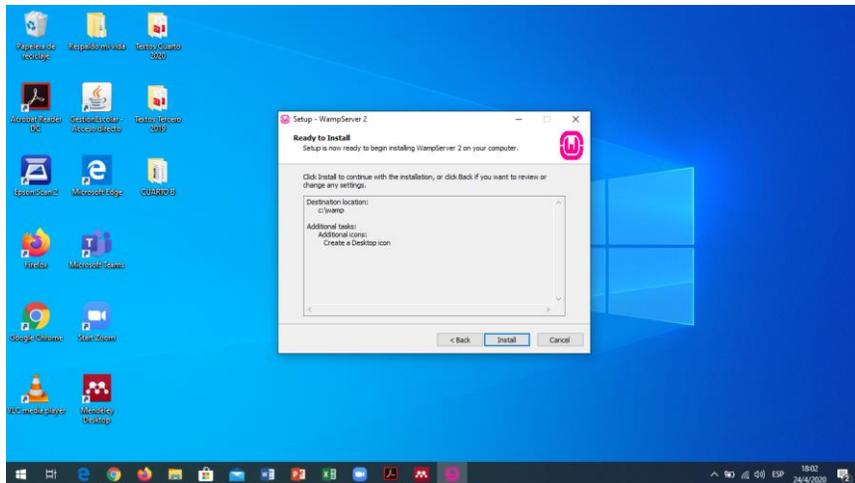


Figura: comenzar la Instalación.
Fuente: Elaborado por el autor.

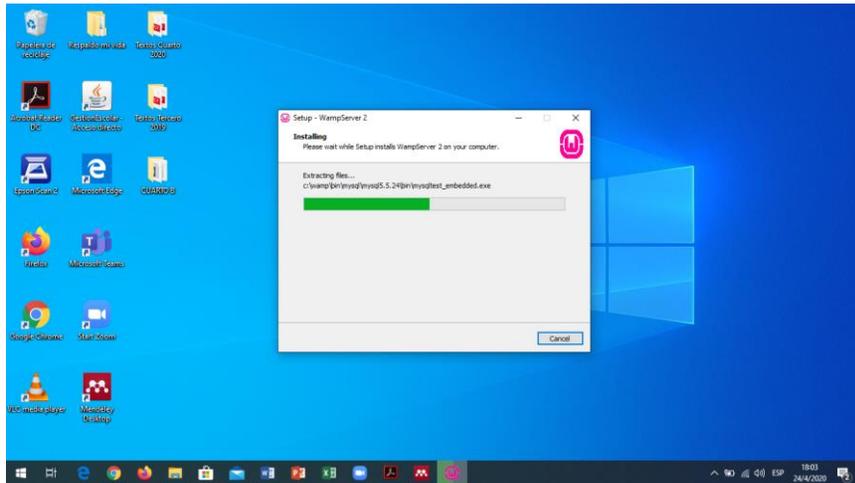


Figura: Wamp server Instalándose.
Fuente: Elaborado por el autor.

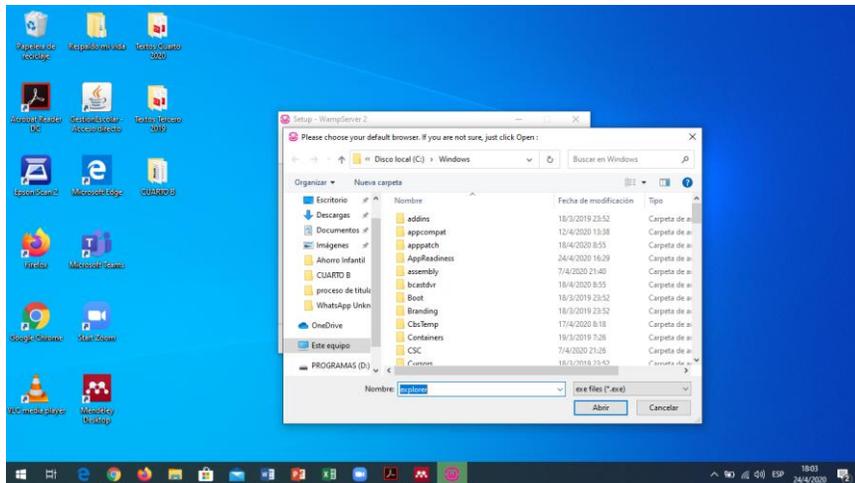


Figura: Seleccionar Explorer y abrir.
Fuente: Elaborado por el autor.

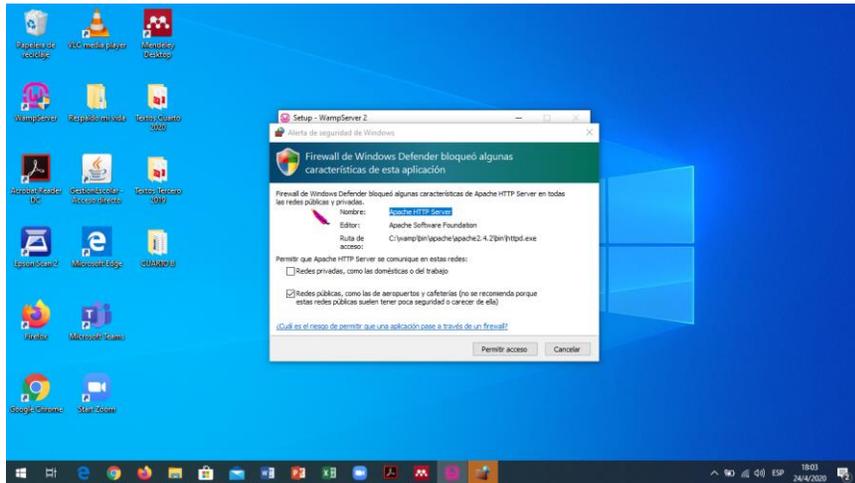


Figura: permitir acceso firewall de Wamp server.
Fuente: Elaborado por el autor.

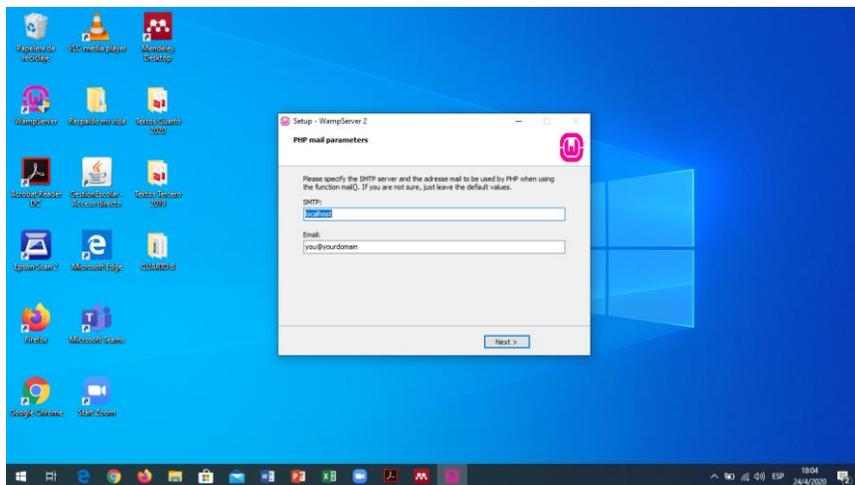


Figura: Parámetros de I-mail de Wamp Server.
Fuente: Elaborado por el autor.

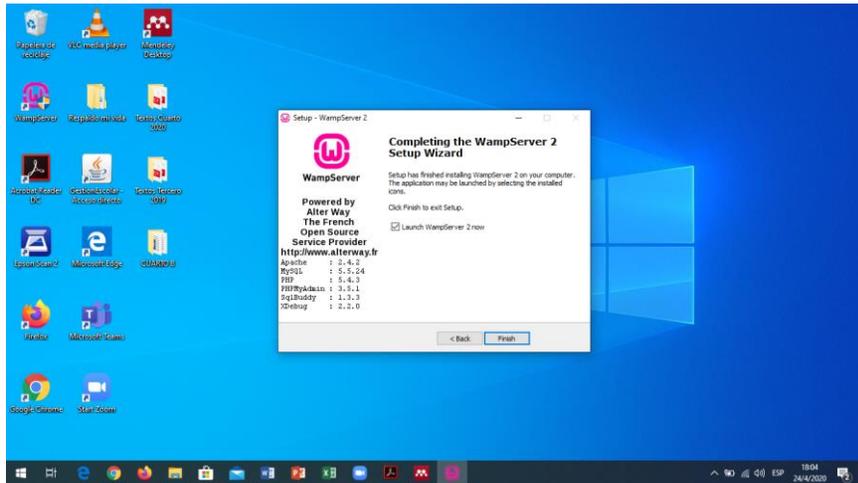


Figura: Fin de la Instalación de Wamp Server.
Fuente: Elaborado por el autor.

2.5.2. Plan de Ejecución.

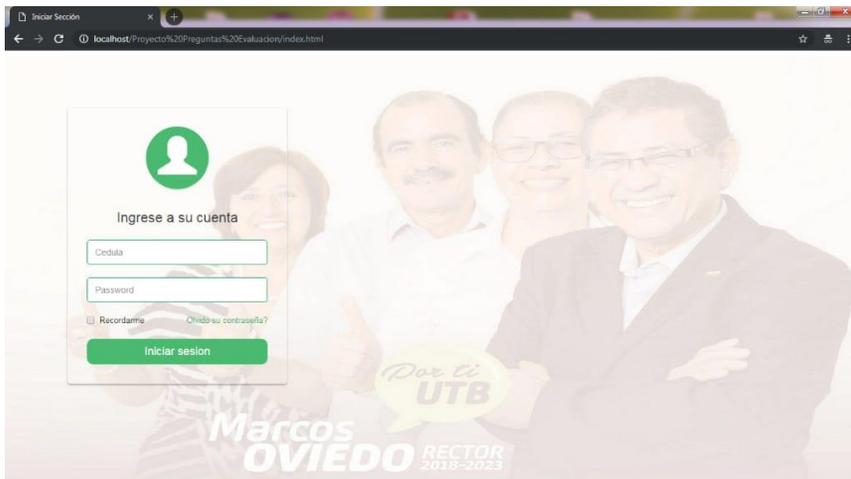


Figura: Página del login del Sistema.
Fuente: Elaborado por el autor.



Figura: Registro de docente encargado.
Fuente: Elaborado por el autor.

Sistema Evaluación

localhost/Proyecto%20Preguntas%20Evaluacion/Insertestudiante.php

Registrar estudiante

* Campos obligatorios

* Codula:

* Nombres:

* Apellidos:

* Dirección:

* Correo:

* Celular:

Guardar Cancelar

Figura: Registro del estudiante.
Fuente: Elaborado por el autor.

Sistema Evaluación

localhost/Proyecto%20Preguntas%20Evaluacion/InsertPregunta.php?id=1

TOMA DEL EXAMEN DE GRADO DE CARÁTER COMPLEXIVO - 2020

Ingresar la pregunta con sus respectivas respuestas:

* Campos obligatorios

* Pregunta:

Respuestas

* A:

* B:

* C:

* D:

Seleccione la Correcta

Guardar Cancelar

Figura: Ingreso de preguntas a la base de datos.
Fuente: Elaborado por el autor.

Registrar la Evaluación

* Campos obligatorios

* Descripción:

* Fecha que se tomará:

* Hora que se tomará:

* Duración que durara:

* Puntaje Máximo:

* Puntaje mínimo:

* N° de preguntas que se tomará:

Figura: Registro de la Evaluación.
Fuente: Elaborado por el autor.

Evaluaciones Registradas

N°	Descripción	Fecha y hora que se tomará	Estado	Activar o Inactivar	Acciones
1	TOMA DEL EXAMEN DE GRADO DE CARÁTER COMPLEXIVO - 2020	2020-06-12 08:00:00	ACTIVO	<input checked="" type="checkbox"/>	<input type="button" value="+"/> <input type="button" value="−"/> <input type="button" value="↺"/> <input type="button" value="↻"/>
2	xxxxxxxxxxxx	2020-03-20 08:00:00	ACTIVO	<input checked="" type="checkbox"/>	<input type="button" value="+"/> <input type="button" value="−"/> <input type="button" value="↺"/> <input type="button" value="↻"/>
3	ódd xxxxxxxxxxx é ñ	2020-03-19 08:00:00	INACTIVO	<input checked="" type="checkbox"/>	

Figura: Evaluaciones Registrada.
Fuente: Elaborado por el autor.

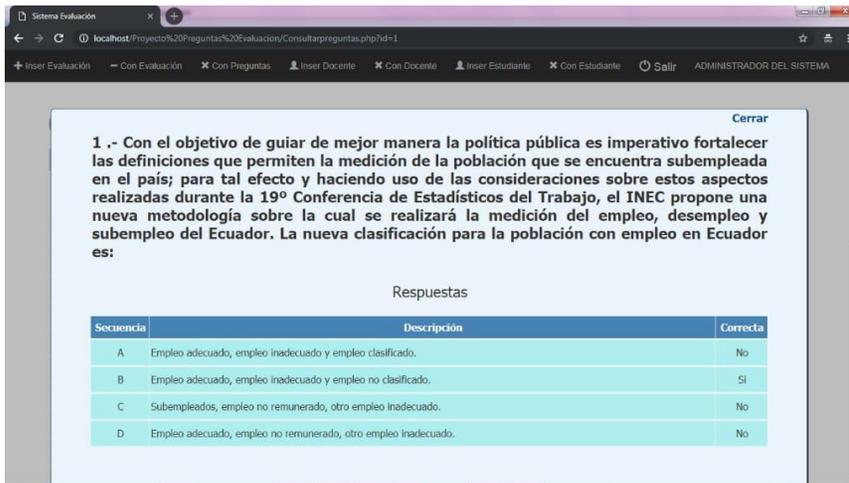


Figura: Evaluaciones Registrada.
Fuente: Elaborado por el autor.

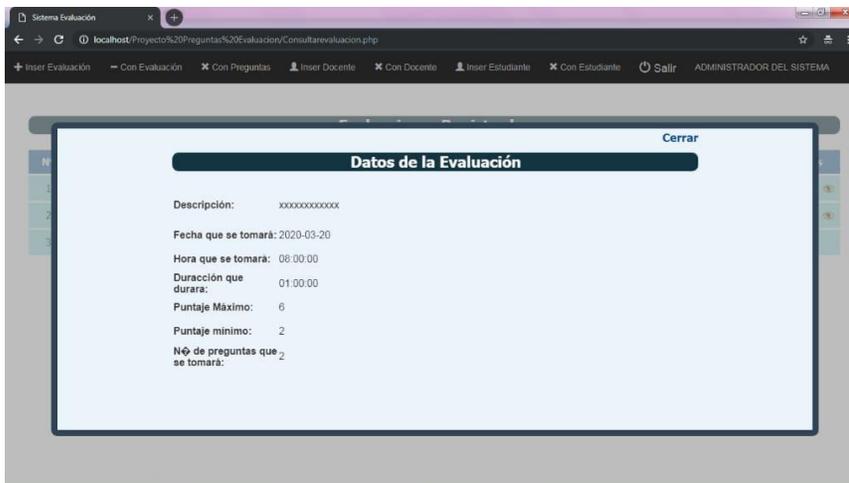


Figura: Datos de Evaluación.
Fuente: Elaborado por el autor.

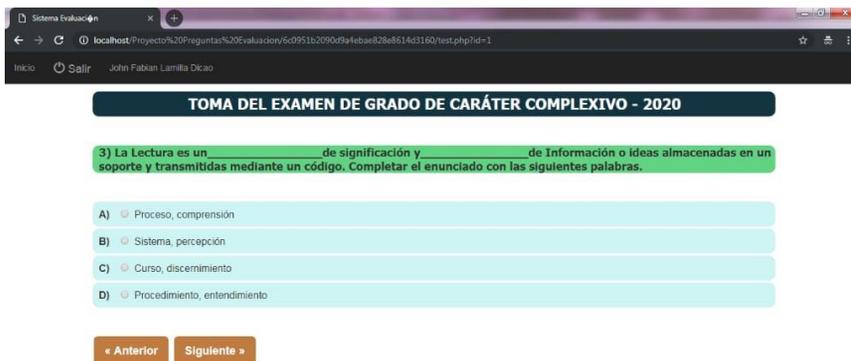


Figura: Evaluaciones Registrada.
Fuente: Elaborado por el autor.

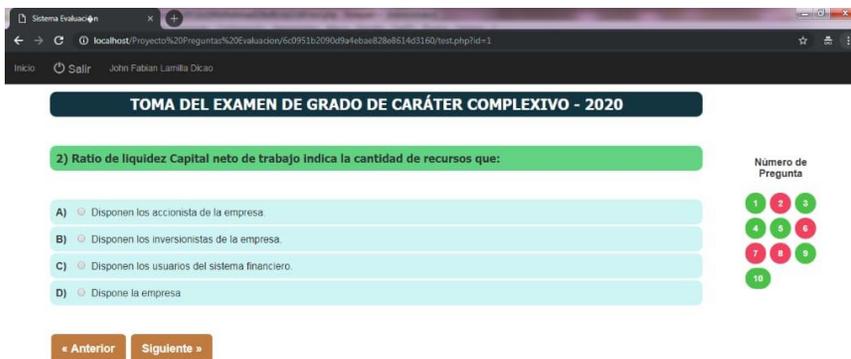
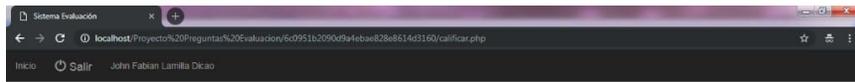


Figura: Datos de Evaluación.
Fuente: Elaborado por el autor.



Resultado de la evaluación TOMA DEL EXAMEN DE GRADO DE CARÁTER COMPLEXIVO - 2020	
Respuestas Correctas	4
Respuestas Incorrectas	6
Puntaje Obtenido	2 / 5

[Salir](#)

Figura: Resultado de la Evaluación.
Fuente: Elaborado por el autor.

CAPITULO III

3. Evaluación del Prototipo.

3.1. Plan de Evaluación.

Para realizar la evaluación del prototipo, se construyó un cronograma de actividades el cual se mostrara el rendimiento del software para la toma de los exámenes complexivos.

Cronograma De Actividades Del Desarrollo Del Prototipo	
Fecha	Actividades
01/04/2020	Instalación del NotePad++, Wamp Server, PostgreSQL.
01/04/2020	Realización de la lluvia de ideas.
01/04/2020	Construir la base datos en un borrador, utilizando los requerimientos del simulador del examen Complexivo.
01/04/2020	Realizar la identificación de las tablas adecuadas del sistema.
01/04/2020	Buscar las relaciones entre la tabla del sistema.
10/04/2020	Reunión con el tutor para definir el modelado correcto del sistema.
11/04/2020	Realización del script de la base datos.

12/04/2020	Normalizar y comprobar las relaciones estén correctas entre las tablas de la base de datos.
12/04/2020	Construir la conexión entre la base datos con el sistema.
13/04/2020	Ver tutoriales de sistemas informáticos realizado en PHP, MySQL y PostgreSQL.
14/04/2020	Realización de login del sistema.
15/04/2020	Diseño de menú inicio del sistema.
16/04/2020	Diseño de la páginas del sistema.
17/04/2020	Diseño de la paginas del sistema
18/04/2020	Diseño de la paginas del sistema
19/04/2020	Diseño de la paginas del sistema
20/04/2020	Diseño de la paginas del sistema
21/04/2020	Diseño de la paginas del sistema
22/04/2020	Diseño de la paginas del sistema
23/04/2020	Diseño de la paginas del sistema
24/04/2020	Realización de la conexión entre todas las páginas del sistema.

24/04/2020	Prueba de registro en algunas tablas.
24/04/2020	Validación de los campos en cada una de las cajas de textos en las páginas.
24/04/2020	Pruebas de prototipo entrando como administrador, Docente y Estudiante.
24/04/2020	Prueba de la caja Blanca
24/04/2020	Prueba de la caja Negra
24/04/2020	Pruebas de funcionamiento del sistema en otro ordenador.

Fuente: Elaborado por el autor.

3.1.1. Prueba de Caja Blanca.

Se derivan en los casos que aseguren que durante la prueba se han ejecutado todas las sentencias del programa al menos una vez y que se ejercitan todas las condiciones lógicas:

Testing de caja blanca: Insertar Estudiantes.

Validación de la caja de texto Cedula.

```

</tr>
<td width="10%">  Cedula:</td>
<td colspan="2">
  <input name="cedula" id="cedula" type="text" maxlength="10" required onKeyPress="return Isnumero(event)"
onBlur="comprobarCedula()">
</td>
</tr>

```

Estamos utilizando Ajax para la validación tiempo real para verificar si la cédula es válida, para eso utilizamos el algoritmo que se aplica en el Ecuador.

Casos de prueba que permita:

- ✓ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ✓ Ejerciten todas las decisiones lógicas en las vertientes verdaderas y falsa.
- ✓ Ejecuten todos los bucles en sus límites operacionales.
- ✓ Ejerciten las estructuras internas de datos para asegurar su validez.

Validación de código de cedula.

```
function validar() {  
var cad = $("#cedula").val().trim();  
var total = 0;  
var longitud = cad.length;  
var longcheck = longitud - 1;  
  
if (cad !== "" && longitud === 10){  
  for(i = 0; i < longcheck; i++){  
    if (i%2 === 0) {  
      var aux = cad.charAt(i) * 2;  
      if (aux > 9) aux -= 9;  
      total += aux;  
    } else {  
      total += parseInt(cad.charAt(i));  
    }  
  }  
}
```

```
total = total % 10 ? 10 - total % 10 : 0;
if (cad.charAt(longitud-1) == total) {
    }else{
    alert("Cedula Inválida");
        $("#cedula").val("");
        $("#cedula").focus();
    }
}
}

function comprobarCedula() {
jQuery.ajax({
url: "Verificardb.php",
data:'cedula='+$("#cedula").val(),
type: "POST",
success:function(data){
    if(data==1)
    {
        alert("Cedula ya Registrada");
        $("#cedula").val("");
        $("#cedula").focus();
    }else
    {
        validar();
    }
},
error:function (){}
});
}
```

3.1.2. Prueba de Caja Negra.

Son pruebas funcionales, tienen como finalidad encontrar aspectos que no estén especificados dentro del contexto del sistema, siendo utilizado también para la revisión del sistema web.

- Entrada y salida de datos por parte de los usuarios.
- Validación de cajas de texto para el ingreso de caracteres específicos.
- Comprobación de que cada una de las páginas se ajuste a los datos necesitados y requeridos.
- Que la información sea mostrada correctamente y entendida por los usuarios.
- Nunca visualizar información innecesaria y secreta en ninguna caso.

The screenshot shows a web browser window with the URL `localhost/Proyecto%20Preguntas%20Evaluacion%3afed0047b08059d0fada10f400c1e5/insertestudiante.php`. The page title is "Sistema pa mplexivo". The main content area is titled "Registrar estudiante" and contains a form with the following fields:

- Cedula:
- Nombres:
- Apellidos:
- Dirección:
- Correo:
- Celular:

There are "Guardar" and "Cancelar" buttons at the bottom of the form. A red asterisk and the text "Campos obligatorios" are visible next to the form fields. The left sidebar contains a navigation menu with the following items:

- Inicio
- Mantenimiento
- Gestión
- Reportes
- Seguridad
- Salir

The footer of the page reads "Copyright © 2020. All Rights Reserved."

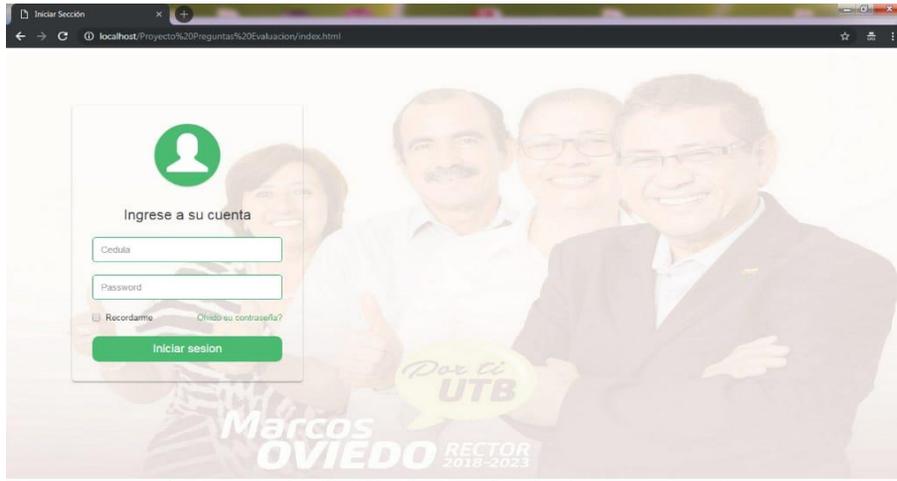


Figura: Verificación de conexión exitosa del Sistema y la Base de Datos.

Fuente: Elaborado por el autor.

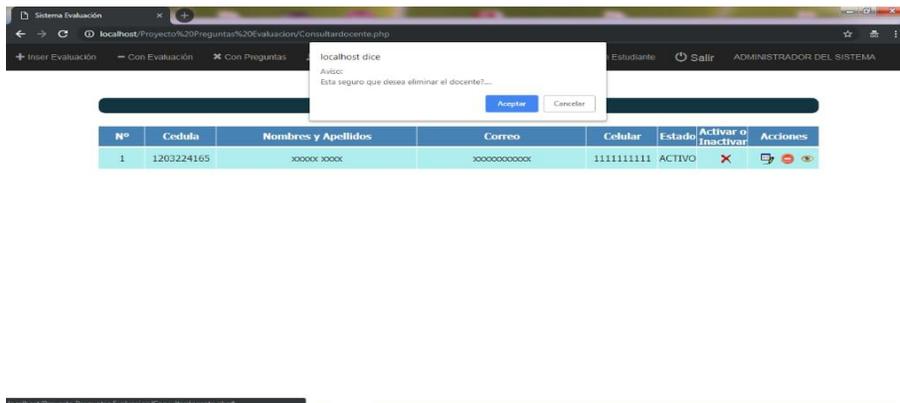


Figura: Inicio de sesión al Sistema como Administrador

Fuente: Elaborado por el autor.

Docentes Registrados							
Nº	Celular	Nombres y Apellidos	Correo	Celular	Estado	Activar o Inactivar	Acciones
1	1203224165	xxxxx xxxxx	xxxxxxx	1111111111	ACTIVO	X	🔄 🗑️

Figura: Verificación de Docentes Registrado en el Sistema.
Fuente: Elaborado por el autor.

3.2.Resultados de la Evaluación.

El sistema aplicativo web desarrollado del simulador para la toma del examen Complexivo, aún no está implementado. Pero en las pruebas realizadas no ha presentado ningún inconveniente al momento de ser ejecutado para la simulación de toma del examen Complexivo.

Datos de la Evaluación	
Descripción:	xxxxxxxxxxxx
Fecha que se tomará:	2020-03-20
Hora que se tomará:	06:00:00
Duración que durara:	01:00:00
Puntaje Máximo:	6
Puntaje mínimo:	2
Nº de preguntas que se tomará:	2

Figura: Resultados de la Prueba de Evaluación.
Fuente: Elaborado por el autor.

CONCLUSIONES

El sistema informático está diseñado en base a la metodología RAD y a la metodología estructurada se tomó algunas herramientas de gran popularidad en el campo del desarrollo de software. Se utilizó UML para visualizar, construir y detallar la documentación del sistema.

Para desarrollar el sistema informático se utilizó el NotePad++ para crear el código PHP, y crear todo el sistema desde cero y realizar cada una de las páginas que han sido necesarias, para que cumpla con todas las funciones.

Al crear esta aplicación web se pretende que la realización para la toma del examen Complexivo sea más ágil y eficaz al momento de efectuarla.

Los estudiantes tendrán un simulador de examen Complexivo que les permitirá medir sus conocimientos y prepararse para el examen oficial de la carrera.

RECOMENDACIONES

Revisar los accesos otorgados a cada perfil, verificar que el archivo de importancia del distributivo.

Actualizar la información de la aplicación en cada periodo revisando los parámetros de las materias por carreras.

Capacitar a los usuarios previos al uso de la aplicación.

Si se va a manipular la base de dato, hay que tener cuidado de no borrar alguna tabla, campo o relación, lo aconsejable es tratar de solo utilizar el sistema informático, por lo que este permite apreciar mejor la información.

BIBLIOGRAFIA

- ❖ .3 *Objetivos específicos de la Gestión de Datos*. (1999). www.epidemiolog.net
- ❖ Armando Canchala. (2011). Analisis de Sistema. *Analisis de Sistemas*, 25.
- ❖ Armando Canchala, P. (n.d.). *UML, ejemplo sencillo sobre Modelado de un Proyecto*.
- ❖ Cesar Krall. (2006). *¿Qué es y para qué sirve UML? Versiones de UML (Lenguaje Unificado de Modelado). Tipos de diagramas UML*. Programas Informaticos. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:i-que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163
- ❖ *Desarrollo y optimización de componentes software para tareas ... - José Luis Villada Romero* - Google Libros. (n.d.). Retrieved April 18, 2020, from https://books.google.com.ec/books?hl=es&lr=&id=pXPnCgAAQBAJ&oi=fnd&pg=PT4&dq=VILLADA,+2015&ots=JKT7_b5Uz2&sig=ScUfnmXmgpcHqQXsOs8STgO8YPw&redir_esc=y#v=onepage&q=VILLADA%2C2015&f=false
- ❖ EcuRed. (2015a). *Pruebas de caja blanca*. https://www.ecured.cu/Pruebas_de_caja_blanca
- ❖ EcuRed. (2015b). *Pruebas de caja negra*. https://www.ecured.cu/Pruebas_de_caja_negra
- ❖ *Gestión de servicios en el sistema informático. IFCT0509 - Ester Chicano Tejada* - Google Libros. (n.d.). Retrieved April 17, 2020, from https://books.google.es/books?hl=es&lr=&id=Oq7KCQAAQBAJ&oi=fnd&pg=PT4&dq=Gestión+de+servicios+en+el+sistema+informático.+IFCT0609&ots=tAFr30kZ_c&sig=2SueM9bal7YQ--JsSDzkXJL01AM#v=onepage&q&f=false
- ❖ Gustavo B. (2019, May 12). *Qué Es WAMP (Guía Para Principiantes)*. Guía de

Principiante. <https://www.hostinger.es/tutoriales/que-es-wamp/>

- ❖ James Martin. (n.d.). *Desarrollo rápido de aplicaciones (RAD): ¿Qué es y como funciona?* Retrieved April 20, 2020, from <https://diagramasuml.com/desarrollo-rapido-de-aplicaciones-rad-que-es-y-como-funciona/>
- ❖ James Rumbaugh, Ivar Jacobson, G. B. (2000). UML(Lenguaje Unificado de Modelado). *Manual de Referencia*.
- ❖ Jhon Neva. (2012, September 22). *análisis estructurado*. Desarrollo. <https://es.calameo.com/read/00168584810801aa4e057>
- ❖ Jimena, H. (2018). *Notepad ++ conceptos*. Conceptos. <https://notepad-plus-plus.org/>
- ❖ Julián Perez Porto, A. G. (2010). *Definición de PHP - Qué es, Significado y Concepto*. Definicion de PHP. <https://definicion.de/php/>
- ❖ Martin, J., & Martin, J. (1990). *Rapid Application Development*.
- ❖ *Prototipado*. (2009). 2009. <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>
- ❖ Reicek. (2014). *Qué es PostgreSQL y cuáles son sus ventajas*. <https://platzi.com/blog/que-es-postgresql/>
- ❖ Wictea. (2015). *¿Qué es el lenguaje de programación PHP? - Base de Conocimientos - ICTEA*. Base de Conocimiento. <http://www.ictea.com/cs/index.php?rp=/knowledgebase/8663/iQue-es-el-lenguaje-de-programacion-PHP.html>