

FACULTAD DE INGENIERIA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

DISERTACIÓN DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA ADMINISTRATIVO PARA UN INSTITUTO DE INGLÉS CASO DE ESTUDIO: LIVING ENGLISH INSTITUTE

ANDERSON IVÁN MERA LAPO

DIRECTOR: ING. EDISON MORA

Quito, enero 2020

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Mera Lapo Anderson Iván portador de la cédula de ciudadanía No. 172061724-8 declaro que los resultados obtenidos y presentados en la investigación que presento como informe final, previo a la obtención del Grado de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido de la investigación, las conclusiones, los efectos legales y académicos que se desprendan del trabajo de investigación y luego de la redacción de este documento serán de propia y exclusiva responsabilidad legal y académica.

Igualmente declaro que todo resultado académico que se desprenda de esta investigación y que sea producto de difusión tendrá como filiación la Pontificia Universidad Católica del Ecuador, reconociendo en la autoría al director del Trabajo de Titulación y otras autoridades que ameritan.

_____

Anderson Iván Mera Lapo

CI. 1720617248

DECLARACIÓN Y AUTORIZACIÓN

Yo, **ANDERSON IVAN MERA LAPO**, con C.I. **172061724-8**, autor del trabajo de graduación intitulado: "**ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA ADMINISTRATIVO PARA UN INSTITUTO DE INGLÉS CASO DE ESTUDIO: LIVING ENGLISH INSTITUTE**", previa a la obtención del grado académico de **INGENIERO DE SISTEMAS Y COMPUTACIÓN** en la Facultad de **INGENIERÍA**:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 19 de diciembre del 2019

Anderson Iván Mera Lapo

C.I. 050314962-7

AGRADECIMIENTO

En primer lugar, doy gracias a Dios por todas las bendiciones recibidas.

Agradecimiento a mis padres por la confianza y el amor entregado, porque sin ellos este logro no hubiese podido ser realidad.

Mis abuelos, José y Dolores por su cariño, amor, valores y educación impartida desde mi niñez hasta mi adolescencia, por motivarme y darme alientos para seguir con mis estudios.

A mis tíos, por saber darme consejos en momentos realmente importantes de mi vida, saber guiarme y ser promotores de superación continua.

Mis hermanos, Melany y Jordy, por su cariño, por ser los pequeños que motivan a día a día dar un paso más, para ser el ejemplo para ellos.

A mi esposa Marivy por su compromiso, apoyo, dedicación y sobre todo por su comprensión en momentos difíciles de este proceso académico y a mi hija Hailey quien es la razón y motivo de seguir luchando cada día más y poder obtener mi título universitario.

Anderson Iván Mera Lapo

DEDICATORIA

Dedico este proyecto a Dios, por la salud, por la bendición de ser un profesional, por darme a todas las personas maravillosas que supieron apoyarme en esta etapa universitaria.

Dedico este proyecto con todo el amor y con toda la emoción del mundo al ser más importante de mi vida, mi madre Emma Lapo, quien me supo brindar todo su apoyo sin importar los obstáculos del camino, por su sacrificio y esfuerzo incansable, por siempre creer en mi capacidad de superar cada obstáculo y por haberme formado como el hijo que soy.

A mis abuelos José y Dolores, por ser los seres más indispensables en mi vida, por brindarme apoyo ante todas las adversidades, por ser esa luz de esperanza en este sueño de ser un profesional, por darme el amor y el cariño en todas mis etapas de vida.

A mis tíos Segundo, Benigno, Fabián y Gladys quienes supieron darme la confianza, quienes me brindaron su mano y me apoyaron en los momentos más difíciles de mi carrera universitaria.

A mis hermanos Jordy y Melany, por ser esa razón de superación, por su amor y su comprensión.

A mi querida esposa Marivy, gracias por creer en mí, por brindarme ese amor, cariño, motivación, por siempre ser mi apoyo y mi consuelo ante los problemas, por día a día seguir soñando en crecer juntos.

A mi amada hija Hailey, el regalo más hermoso que Dios me ha regalado, quien es mi inspiración y mi motivación para poder superarme cada día más y poder brindarte todas las oportunidades en su vida.

Anderson Iván Mera Lapo

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA ADMINISTRATIVO PARA UN INSTITUTO DE INGLÉS CASO DE ESTUDIO: LIVING ENGLISH INSTITUTE

RESUMEN

Este trabajo de titulación implementa un sistema automatizado para el control de estudiantes del instituto LIVING ENGLISH en la ciudad de Santo Domingo. El sistema tiene la capacidad de agilizar los procesos manuales que realiza la institución con el objetivo de mejorar las actividades realizadas por estudiantes, docentes y tareas administrativas como matricula, facturación, asistencias de tal manera que la calidad de la atención se racionalice y la productividad aumente.

Se utilizó el marco de desarrollo RUP (Rational Unified Process) el cual es una metodología orientada a objetos, para la recopilación de información y funcionalidad del sistema se empleó la entrevista. En el desarrollo y la implementación se utilizó herramientas de software libre como PHP v. 7.3.10 dentro del framework Laravel versión 6.0 y Javascript a partir del framework VUEjs versión 2.7, para el almacenamiento de datos se utilizó el sistema gestor de base de datos MySQL.

El sistema brinda una interfaz gráfica (UI) estructurada dentro de un dashboard para que vaya de acorde a la experiencia del usuario (UX). El sistema agilizara los procesos implementados dada la escalabilidad, flexibilidad y alto rendimiento de la base de datos.

Palabras Clave: Software, RUP, Laravel, VUEjs, Tecnología, Interfaz de Usuario, Experiencia de Usuario, Sistema Administrativo.

ABSTRACT

This degree work implements an automated system for the control of students of the LIVING ENGLISH institute in the city of Santo Domingo. The system has the ability to streamline the manual processes carried out by the institution in order to improve the activities carried out by students, teachers and administrative tasks such as enrollment, billing, assistance in such a way that the quality of care is rationalized and productivity increases .

The RUP (Rational Unified Process) development framework was used, which is an object-oriented methodology, for the collection of information and system functionality the interview was used. In the development and implementation, free software tools such as PHP v. 7.3.10 within the Laravel version 6.0 and Javascript framework from the VUEjs version 2.7 framework, the MySQL database management system was used for data storage.

The system provides a graphical interface (UI) structured within a dashboard to match the user experience (UX). The system will streamline the processes implemented given the scalability, flexibility and high performance of the database.

Keywords: Software, RUP, Laravel, VUEjs, Technology, User Interface, User Experience, Administrative System.

ÍNDICE DE CONTENIDO

RESUMEN 7
ABSTRACT 8
1. INTRODUCCIÓN 13
1.1. Antecedentes 13
1.2. Datos de la organización o institución 14
1.3. Justificación 14
1.4. Objetivos de la Implementación 15
1.4.1. Objetivo General 15
1.4.2. Objetivos Específicos 15
1.5. Alcance 15
2. ESTADO DEL ARTE 16
3. PROCESO DE DESARROLLO DE SOFTWARE RUP 19
3.1. Procesos internos de la metodología 19
3.1.1. Proceso Unificado dirigido por casos de uso 19
3.1.2. Proceso unificado centrado en la arquitectura 19
3.1.3. Proceso unificado es iterativo e incremental 20
3.2. Fases de la Metodología 20
3.2.1. Inicio 21
3.2.2. Elaboración 22
3.2.3. Construcción 23
3.2.4. Transición 23
3.2.5. Roles 24
4. ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA 25
4.1. Detalle funcional en la gestión de estudiantes 27
4.2. Detalle funcional en la gestión de docentes 27
4.3. Detalle funcional en la gestión de asistencias 28
4.4. Detalle funcional en la gestión de horarios 29
4.5. Detalle funcional en la gestión de matriculas 30
4.6. Detalle funcional en la gestión de facturación 31
4.7. Detalle funcional en la gestión de niveles 31
4.8. Proceso manual de matrícula 32
4.9. Proceso manual de horarios 34
4.10. Proceso manual de Selección de Horarios del Estudiante 35
4.11. Proceso manual de facturación 36
5. DESARROLLO DE LAS FUNCIONES DEL SISTEMA 37

5.1. Proceso de matrícula 37
5.2. Proceso de definición de horarios 39
5.6. DIAGRAMAS DE SECUENCIA 43
6. IMPLEMENTACIÓN DEL SISTEMA 54
6.1. Modelado de la base de datos 54
6.2. Herramientas e integración de procesos en sistema 57
6.3. Pruebas de validación 59
6.4. Configuración del proyecto en el Hosting 60
6.5. Discusión 61
7. CONCLUSIONES 63
8. RECOMENDACIONES 64
BIBLIOGRAFÍA 65
9. GLOSARIO 67

ÍNDICE DE FIGURAS

Figura 1. Fases de la metodología RUP y sus flujos de trabajo 21
Figura 2. Diagrama general de las funcionalidades del sistema administrativo obtenida del levantamiento de requerimientos. 26
Figura 3. Diagrama funcional de la administración de estudiantes con sus debidos procesos de: ingreso, edición, búsqueda y eliminación lógica. 27
Figura 4. Diagrama especifico funcional de la administración de docentes con sus respectivas acciones: ingresas, editar, eliminar y realizar búsquedas. 28
Figura 5. Diagrama funcional de la administración de asistencias, permite agregar asistencias, actualizarlas y realizar búsquedas de un estudiante para emitir un reporte de asistencias. 28
Figura 6. Diagrama funcional de la administración de horarios, en ella se detallan los procesos de creación individual: registro, edición, eliminación lógica y consultas de manera general para cursos anuales, mientras tanto la creación personalizada parte del estudiante que lo solicita. ... 29
Figura 7. Diagrama funcional de la administración de matrículas, se plantea el registro de matriculas a nuevos estudiantes o estudiantes de reingreso al instituto. 30
Figura 8. Diagrama funcional de la administración de facturación para el ingreso, anulación y emisión de reportes por cada usuario registrado. 31
Figura 9. Diagrama funcional de la administración de niveles la funcionalidad debe permitir: registrar, editar, eliminar y consultar por parámetros un nivel. 31
Figura 10. Detalle del proceso manual de matriculación, en el registro de un estudiante. 33
Figura 11. Proceso manual para el registro de horarios, dependiendo del tipo de programa seleccionado. 34
Figura 12. Proceso manual de selección horario por el estudiante. 35
Figura 13. Proceso manual de facturación dentro del instituto. 36
Figura 14. Proceso automatizado de matrícula. 38
Figura 15. Proceso de definición de horarios, con periodo de tiempos de uso y encargados de área. 39
Figura 16. Proceso de selección horario estudiantes, con detalle de tiempo en uso y encargados. 40
Figura 17. Proceso de control de facturación, detalle de tiempos de uso y personal encargado. 41
Figura 18. Proceso de control de asistencias, detalle de tiempos de uso y personal encargado. 42
Figura 19. Diagramas de secuencias de estudiantes caso de uso estudiante: secuencia de forma correcta. 43
Figura 20. Caso de uso estudiante: se considera incorrecto cuando existe datos incompletos o campos duplicados. 44
Figura 21. Caso de uso estudiante: datos existentes en la base de datos. 44
Figura 22. Diagramas de secuencias de docentes caso de uso docente: ingreso correcto de datos. 45
Figura 23. Caso de uso docente: ingreso incorrecto. 46
Figura 24. Diagramas de secuencias de docentes caso de uso docente: existente. 46
Figura 25. Diagramas de secuencias de asistencias caso de uso asistencia: correcto. 47
Figura 26. Diagramas de secuencias de asistencias caso de uso asistencia: incorrecto. 47
Figura 27. Diagramas de secuencias de horarios caso de uso horarios: correcto. 48
Figura 28. Diagramas de secuencias de horarios caso de uso horarios: incorrecto. 48
Figura 29. Diagramas de secuencias de facturación caso de uso factura: correcto. 49
Figura 30. Diagramas de secuencias de facturación caso de uso factura: incorrecto. 50
Figura 31. Caso de uso horarios estudiantes: correcto. 51
Figura 32. Caso de uso horarios estudiantes: incorrecto. 52

Figura 33. Diagramas de secuencias de niveles caso de uso niveles: correcto. 53
Figura 34. Diagramas de secuencias de niveles caso de uso niveles: datos incorrectos o campos vacíos..... 53
Figura 35. Modelo conceptual de la base de datos: implementación de las entidades sin atributos y claves primarias..... 55
Figura 36. Modelo físico de la base de datos: implementación de atributos y claves primarias 55
Figura 37. Modelo entidad relación: representada con las relaciones que intervienen en el proceso administrativo y gestión de procesos. 56
Figura 38. Login de Acceso al sistema: solicita usuario y contraseña para ingresar. 58
Figura 39. Sistema administrativo: se presenta cada uno de los módulos que responden a los procesos desarrollados..... 58
Figura 40. Listado del registro de facturación en el sistema. 59
Figura 41. Proceso de facturación: se detalla el nombre del estudiante, el tipo de comprobante, la serie de comprobante, el número de comprobante, la matricula, el precio y la cantidad de matrículas a pagar. 59
Figura 42. Pruebas de validación PHP con respuesta del servidor code:200. 60
Figura 43. Herramienta de pruebas VUEJS: validación de métodos y atributos procesados de manera correcta. 60
Figura 44. Integración del sistema: Dominio de acceso especificado como livingenglishinstitute.ec..... 61

ÍNDICE DE TABLAS

Tabla 1. Personal del instituto de inglés: gerente, administrador, RRHH, y Administrador de TI. 25

1. INTRODUCCIÓN

Actualmente en todas las empresas existe la manipulación del Internet, los procesos manuales cada vez son menos necesarios para producir servicios o diferentes actividades, debido a estas razones surge la necesidad de crear aplicativos webs, los cuales permiten administrar por medio de una conexión a Internet una empresa o un negocio.

El instituto LIVING ENGLISH realiza procesos manuales que no soportan la demanda de estudiantes en cada programa que presenta, entonces surge la necesidad de implementar un sistema automatizado que le brinde una mayor agilidad y eficiencia durante los procesos administrativos.

Las limitaciones que presenta actualmente la empresa son: la mala calidad y la falta de gestión de información, con la integración del sistema se podrá reducir los tiempos de atención e implementar tecnologías modernas para administrar el instituto.

Con el desarrollo del sistema y la integración en la institución se promueve el uso de nuevas tecnologías para brindar una educación de calidad y evitar la pérdida de información, manteniendo la información en una base de datos y no almacenar registros en carpetas.

1.1. Antecedentes

La misión del instituto Living English es capacitar y enseñar responsablemente a sus clientes el lenguaje inglés superando siempre sus expectativas, personales y profesionales.

La visión que presenta es ser el referente en el campo de la enseñanza y aprendizaje del idioma ingles de niños, jóvenes y adultos a nivel regional.

Los valores corporativos son: Servicio, Responsabilidad, Solidaridad, Conducta Ética, Servicio Público.

Existen varios sistemas integrados para instituciones de inglés, en los cuales se establece la administración de la empresa, debido a la alta demanda de estudiantes, se registran a los estudiantes, los pagos, los niveles a los que pertenecen y los horarios que estos

presentan clases. Un sistema integrado para el aprendizaje y control del área administrativa.

Los sistemas web para instituciones educativas son de gran ayuda y necesarios para implantar una adecuada gestión de procesos y no presentar perdidas de documentación o mala calidad de servicio; en el presente caso se necesita implementar un sistema web para el control de estudiantes con el objetivo de aumentar la eficiencia, rapidez y evitar la pérdida de información.

1.2. Datos de la organización o institución

Nombre de la institución: LIVING ENGLISH INSTITUTE

Actividad: Es una entidad educativa con más 10 años en el mercado en la provincia de Santo Domingo, dedicada a la enseñanza integral del idioma inglés, con instructores altamente capacitados y con vasta experiencia educativa.

Ubicación: Santo Domingo de los Tsáchilas - Av. Abraham Calazacón y pasaje La Paz - Código Postal 230103

Características: La institución se encuentra ubicada estratégicamente en la zona céntrica – comercial de la urbe, cuenta con instalaciones debidamente equipadas en audio y video, dividida en aulas por niveles y edades: Little kids, Kids, Big Kids, Teen and Teen Advance, en donde impartimos clases prácticas, didácticas y dinámicas.

1.3. Justificación

El instituto de inglés presenta la necesidad de un aplicativo web que ayude a gestionar la administración del instituto, busca satisfacer el nivel competitivo con respecto a la competencia.

Al implementar este tipo de sistema se brindará una agilidad de procesos donde el usuario no necesite estar por tiempos muy prolongados en espera para ser atendido o realizar algún tipo de tramite dentro de la institución.

La gestión administrativa será capaz de controlar los reportes mensuales de sus ingresos, estudiantes, matriculas, facturas y será posible mantenerse notificado en tiempo real de los procesos realizados en el día.

1.4. Objetivos de la Implementación

1.4.1. Objetivo General

Analizar, Desarrollar e Implementar un Sistema Administrativo para el Instituto de Inglés “Living English Institute”.

1.4.2. Objetivos Específicos

Analizar e incorporar los factores que afectan a la creación de la base de datos.

Desarrollar un sistema automatizado para gestionar los datos de los estudiantes y del instituto.

Implementar el sistema en el hosting y servidor de la empresa.

1.5. Alcance

El alcance de esta disertación contempla el análisis, desarrollo e implementación de un sistema administrativo para el Instituto Living English Institute. Este debe poder ser usado por el director, administrador y encargados de área perteneciente al instituto. Además, los usuarios en general tendrán acceso solo a información de promociones o avisos.

Al implementar el sistema administrativo se podrá evitar la falta de control y pérdida de datos. El sistema debe controlar el ingreso de datos logrando llevar un control ordenado y unificado para presentar informes mensuales.

2. ESTADO DEL ARTE

Los sistemas transaccionales para la gestión de la información cada vez se tornan más indispensables, dado que es necesario crear soluciones automatizadas para satisfacer una problemática definida.

Actualmente, los SI se enfrentan a dos retos fundamentales. En primer lugar, su diseño, desarrollo e implementación son procesos que confluyen en diferentes contextos, con distintos puntos de vista y suposiciones acerca de determinado dominio. Esto provoca problemas de comunicación por falta de entendimiento compartido y por la complejidad de la realidad. En segundo lugar, las representaciones en los SI deben corresponderse, lo más estrechamente posible, con la realidad y los procesos que ellos representan para que finalmente cumplan con los objetivos diseñados. (Díaz Pérez M, 2009)

Cuando surge el requisito de automatizar un proceso o conjuntos de actividades de una empresa, se debe analizar cuáles son los funcionalidades requeridas y a partir de ellas poder crear una solución. Dando lugar al desarrollo de un sistema transaccional que permitirá la administración de la información de una manera organizada y estructurada.

Así mismo, los sistemas académicos son integraciones de los sistemas transaccionales, pero en la rama de la educación, dado que las instituciones parte de un sistema transaccional para poder administrar una empresa, universidad o instituto. Cabe recalcar que la educación ha sido un proceso fundamental para el desarrollo de los diferentes grupos humanos. Debido a esto surge la necesidad de implementar un sistema automatizado en las instituciones educativas, donde se debe brindar la facilidad para gestionar la información de un estudiante.

Los sistemas web académicos cumplen con ese rol, satisfacen la necesidad de administrar los datos de una universidad o instituto, sin embargo, existen varias alternativas a la hora de seleccionar una herramienta para soportar dicha gestión académica.

Entre las principales herramientas de gestión de información educativa se presenta: **Additio App, aGora, Alexia, ApliAula, Aula1.** Se ha seleccionado las diferentes herramientas web que brindan la oportunidad de satisfacer el control de una institución o el desarrollo académico. Cada una de las aplicaciones anteriores cumplen papeles diferentes para su comparativa se detallará brevemente cada una.

Additio App

Es una herramienta web y disponible para dispositivos Android y iOS, permita realizar la planificación del docente, el control de asistencias, control de notas, entre sus principales características incluyen: creación de tutorías personalizadas, personalizar horarios, observar las tareas a realizar, realizar evaluaciones y permite trabajar sin conexión a internet.

aGora

Es un aplicativo web que cubre con las necesidades de un centro académico con la planificación, gestión y control académico, comercial y económico. Es una herramienta que cuenta con respaldos de usuarios, su característica es que es escalable, potente y de rápida ejecución.

Alexia

Software totalmente web, cuenta con más de 1200 centros educativos, debido que aporta flexibilidad en las áreas de facturación, impagados, ventas, contabilidad, evaluación, programación de aulas, seguimiento del alumno, se integra con office 365 y Moodle.

ApliAula

Brinda la gestión de mensajería, ausencias de docentes o estudiantes, documentos, notas, exámenes, evaluaciones, inscripciones, mantiene la comunicación entre docente y representante.

Aula1

Gira a un entorno virtual de aprendizaje, genera horarios, presenta un cuaderno digital, emails anti spam, repositorio de documentos propios, gestión de docentes y administración.

Al conocer cada una de las herramientas más usadas en el mercado por las diferentes instituciones, se ha planteado la necesidad de desarrollar un aplicativo web que brinde las principales características de cada herramienta anteriormente detallada. Para esto se ha seleccionado un instituto de inglés, centro educativo que administrara información de docentes, estudiantes, horarios, asistencias, calificaciones, documentos, reportes mensajería instantánea, matricula, notificaciones en tiempo real y permisos para gestionar pagos previo a cancelación de curso.

El desarrollo de esta herramienta permite que la institución brinde un servicio orientado a la educación tecnología debido que integrando un sistema educativo-administrativo permite conocer cómo trabaja la institución y que tipo de servicios brinda. La implementación de esta herramienta tiene como objetivo brindar un entorno virtual para poder administrar la información interna del instituto y a su vez permitir la relación entre el representante y la institución.

Finalmente cabe recalcar que la herramienta a desarrollar cumplirá con las necesidades primordiales de la institución, pero a su vez se presenta la brecha de escalabilidad para futuros cambios o creación de nuevos módulos.

3. PROCESO DE DESARROLLO DE SOFTWARE RUP

El Proceso Unificado proporciona un poderoso marco de trabajo para el desarrollo de aplicaciones, el cual identifica las actividades necesarias y ayuda a la disposición de un plan oficial para el proceso de desarrollo de software. RUP brinda una metodología exacta, completa, adaptable al contexto y a las necesidades de una organización, además tiene como principal meta asegurar la producción de un software de alta calidad.

Se implementó la metodología RUP debido que es necesario realizar un análisis detallado de todos los requerimientos para comprender su complejidad. Al ser un proceso rígido e inmodificable asienta sus bases en el desarrollo de componentes los cuales deben ser aprobados por el diseñador de la solución para seguir con el desarrollo del sistema.

3.1. Procesos internos de la metodología

Se define a los procesos internos como el pilar para dar cumplimiento a cada una de las fases establecidas dentro de esta metodología, estas son establecidas de manera secuencial con lo que se garantiza un producto de calidad que es entregado al usuario final. A continuación, se detalla cada uno de estos en su respectivo orden.

3.1.1. Proceso Unificado dirigido por casos de uso

Describe el servicio que ha sido presentado por un usuario u otro actor externo, este detalla la funcionalidad del sistema y presenta la secuencia completa de iteraciones entre el actor y el sistema.

Se define como un caso de uso a un fragmento funcional el cual permite conocer un resultado de la necesidad del usuario, al mismo tiempo que guía su diseño, implementación y pruebas (Jacobson, Booch, & Rumbaugh, 2000). Los casos de uso no se desarrollan aisladamente, es decir que estos guían la arquitectura del sistema dado que permite comprender lo que realizará el software (Jacobson, Booch, & Rumbaugh, 2000).

3.1.2. Proceso unificado centrado en la arquitectura

Ilustra las diferentes vistas del sistema a desarrollar, la arquitectura del software es importante para comprender el sistema como un todo y a la vez en sus distintas partes.

La arquitectura se presenta de acuerdo con las necesidades de la empresa, además como lo perciben los usuarios se debe reflejar en los casos de uso. Sin embargo, también depende de factores como la plataforma en donde será implementado para su funcionamiento.

Por tanto, una arquitectura se debe presentar paralela a los casos de uso para permitir que el sistema evolucione, no solo en su desarrollo inicial, sino también a lo largo de futuras modificaciones (Jacobson, Booch, & Rumbaugh, 2000).

El uso del proceso centrado en la arquitectura sirve para organizar el desarrollo, fomentar la reutilización de componentes y hacerlo evolucionar, por último, permite agregar nuevas funcionalidades, generando aplicativos flexibles y escalables.

3.1.3. Proceso unificado es iterativo e incremental

El proceso es iterativo e incremental dado que se divide en pequeños proyectos, donde cada uno cumple con sus especificaciones; El desarrollo es una iteración que va incrementando la funcionalidad del sistema.

Para desarrollar cada mini proyecto se basan en dos factores, el primero identifica un grupo de casos de uso que amplían la funcionalidad del sistema y el segundo en los riesgos más importantes durante la implementación.

Por supuesto, un incremento no debe ser necesariamente aditivo debido que en las primeras fases del desarrollo, los desarrolladores pueden tener que reemplazar un diseño superficial por uno más detallado o sofisticado, sin embargo en las siguientes fases los incrementos son aditivos (Jacobson, Booch, & Rumbaugh, 2000, pág. 7).

El beneficio de desarrollar el sistema con iteraciones es la priorización de los requerimientos por parte del cliente y del desarrollador. Al evaluar el nivel de criticidad y el riesgo de cada uno, se genera una lista con la ruta crítica del proyecto.

3.2. Fases de la Metodología

La estructura del ciclo de vida de la metodología RUP se divide en 4 fases: inicio, elaboración, construcción y transición, así mismo cada fase tiene su flujo de trabajo. Con ello luego de cada iteración los directores deben tomar las decisiones de aceptación para

continuar con la siguiente fase, por lo tanto, se presentan las fases y los flujos de trabajo a continuación en la figura1.

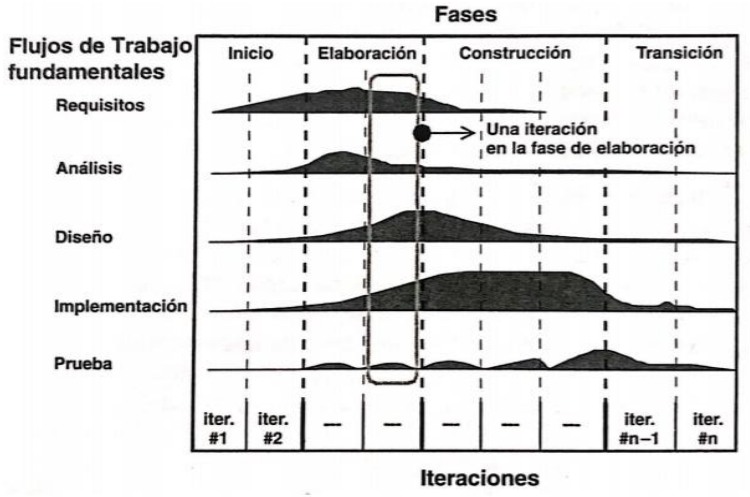


Figura 1. Fases de la metodología RUP y sus flujos de trabajo
Fuente: (Jacobson, Booch, & Rumbaugh, 2000)

3.2.1. Inicio

Durante fase de inicio se describe los flujos de trabajo necesarios para el acuerdo de las partes interesadas conformada por los objetivos, la arquitectura y la planificación del proyecto.

En su primer aporte se levanta los requerimientos esenciales del sistema, los cuales se transformarán en los casos de uso, además se realiza una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el sistema (Jacobson, Booch, & Rumbaugh, 2000).

Se elabora cuáles serán las limitaciones que tendrá el proyecto, sus procesos y los escenarios que presenta durante el desarrollo del aplicativo, se estiman costos de recursos, tiempos de implementación, riesgos y problemas de incertidumbre a cubrir en el transcurso del desarrollo.

Se debe reunir la información antes del inicio del proyecto, organizarla y descubrir lo que falta no en términos de las fases, sino en términos de los objetivos limitados. Así mismo se traza una línea precisa alrededor de lo que ha de estar dentro y fuera del sistema propuesto, se debe identificar cuáles son los actores que van a interactuar con el producto final (Jacobson, Booch, & Rumbaugh, 2000).

Los criterios para evaluarse son la claridad del sistema, la identificación de los actores, la naturaleza de las interfaces con los actores.

Una de las características principales es considerar las tecnologías sobre la cual va a ser construida, para evidenciar si es eficiente y permite optimizar los recursos existentes, se debe agregar que es tolerante a fallos a su vez flexible a cambios y facilidad de escalabilidad.

Como resultado de integrar todas las actividades y procesos en esta etapa se obtiene un producto con la lista de características más aun una primera versión del modelo de negocio, un esbozo de los modelos de casos de uso, el modelo de análisis y un prototipo del sistema (Jacobson, Booch, & Rumbaugh, 2000).

3.2.2. Elaboración

Se construye sobre el trabajo de la fase de inicio, dado que en la fase de elaboración se desarrollan la mayoría de los casos de uso estimados del producto y se diseña la arquitectura del sistema. Se debe mantener una relación primordial entre la arquitectura del sistema y el producto final (Jacobson, Booch, & Rumbaugh, 2000).

El objetivo de la fase de elaboración es reducir los riesgos a un nivel aceptable para la fase de construcción, dado que aborda los riesgos técnicos que tienen carácter en la arquitectura.

Durante la elaboración se tiene la oportunidad de iterar y ensayar diferentes soluciones como pueden ser usos de tecnología, el marco de trabajo, las estructuras, el diseño corporativo.

El resultado de la fase de elaboración es una línea base de la arquitectura la cual es lo suficientemente robusta para resistir la fase de construcción y la adición de características en posibles cambios del sistema.

Al final de la fase de elaboración, el director del proyecto está en disposición de planificar las actividades y estimar los recursos necesarios para terminar el proyecto (Jacobson, Booch, & Rumbaugh, 2000, pág. 11).

3.2.3. Construcción

El desarrollo de la fase de construcción empieza a partir de la línea base de la arquitectura y trabaja a través de las iteraciones e incrementos para desarrollar un producto con su demostración de operación inicial además debe cumplir con las necesidades del usuario.

Se debe agregar que inicia el desarrollo físico del software, códigos de producción, puesto que se debe tener presente que la capacidad operativa está suficientemente madura y estable para permitir la entrega de versiones (Jacobson, Booch, & Rumbaugh, 2000).

Cada construcción de los componentes implementa un conjunto de casos de uso, además los criterios de evaluación se basan en los requisitos funcionales y no funcionales. Teniendo en cuenta que el resultado de la fase es haber realizado la integración del sistema, las cuales a medida que transcurren las pruebas de integración se comprueba si los resultados son iguales que los objetivos planteados.

Por otra parte, después de validar los resultados y verificar su implementación se procede a revisar lo logrado en las iteraciones contrastándolo con lo planificado. Es necesario recalcar que luego de la integración se debe actualizar los riesgos para determinar que el producto ha superado las pruebas del sistema y que ha logrado la capacidad operativa.

Al final de la fase de construcción se autoriza la fase de transición, por lo que se refiere a que la planificación no debe ser detallada debido a que se realizarán pruebas alfa, pruebas beta que se llevarán a cabo en la fase de transición en usuarios seleccionados (Jacobson, Booch, & Rumbaugh, 2000).

3.2.4. Transición

Durante el desarrollo de transición se pretender realizar las pruebas en usuarios seleccionados, las pruebas beta se deben realizar para comprobar que todo el proceso ha sido cumplido juntamente con los objetivos planteados (Jacobson, Booch, & Rumbaugh, 2000).

Los objetivos deben cumplir con todos los requerimientos que fueron establecidos en etapas anteriores, hasta alcanzar la aprobación de todos los usuarios (Jacobson, Booch, & Rumbaugh, 2000).

Durante el desarrollo se debe implementar el producto entorno al medio donde estará en producción, una vez implementado se analiza si cumple las demandas de los usuarios y del negocio. Se debe interactuar con el sistema para encontrar posibles errores que fueron inesperados en la implementación y durante la fase de construcción. Todavía cabe recalcar que los resultado obtenidos serán analizados para comprender si la iteración cumplió con los requerimientos del sistema ya que se realizaron pruebas alfa y beta las cuales definen si el proyecto esta culminado o necesita modificaciones.

3.2.5. Roles

Los roles son las responsabilidades que han sido asignadas a todo el equipo de desarrollo del sistema, cada uno de ellos es responsable de llevar a cabo los procesos necesarios en cada componente designado para su implementación. Se debe mencionar que existen varios encargados del proyecto, entre ellos están los analistas de requerimientos, el jefe de desarrollo y el jefe de pruebas. Cada uno de ellos se ve ligado en cada fase, dado que si existen problemas o errores en la implementación se debe modificar en una segunda iteración.

Luego de haber realizado cada uno de ellos sus debidas obligaciones se organizan para integrar cada componente y entregar un producto para sus pruebas respectivas en la fase de transición.

4. ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA

A continuación, se realizará el análisis de los requerimientos del sistema de administración a implementarse en el instituto de inglés, además se desarrollará cada requerimiento solicitado por parte de la institución de acuerdo con la metodología determinada.

Durante el análisis y levantamiento de requerimientos se detallará cuáles son las principales necesidades de los usuarios, los problemas de administración y las causas de la baja productividad. Se partirá de la recolección de ideas para proceder a representarlas en casos de uso dando como resultado la creación de la arquitectura para cumplir con objetivos planteados. Por esta razón los integrantes de la institución detallaran las principales necesidades en sus áreas de producción, esto permitirá obtener una amplia visión del sistema en general.

Dentro de la institución se ha recolectado información de los actores que intervienen en el proceso administrativo, véase la tabla 1.

No.	**Nombre**	**Cargo**
1	Jorge Loyola	Gerente y propietario
2	Yadira Jaramillo	Administradora
3	Margarita Loyola	Recursos Humanos
4	Anderson Mera	Ingeniero de desarrollo y soporte técnico

Tabla 1. Personal del instituto de inglés: gerente, administrador, RRHH, y Administrador de TI.
Fuente: Living English Institute

Luego de observar los actores que participan dentro de la institución, se realizó la debida entrevista para recolectar la información necesaria por cada usuario, por consiguiente, cada actor coincide en la falta de un sistema automatizado que brinde las herramientas necesarias para agilizar los procesos administrativos.

Debido a la falta de agilidad y productividad de parte de la institución se plantea desarrollar una solución automatizada que brinde la facilidad de ejecución de los procesos internos de la empresa.

Luego de recopilar la información se debe organizar los requerimientos descritos de acuerdo con el orden de importancia en su ejecución como son las siguientes funcionalidades representadas con la abreviación de F1, F2, Fn: la gestión de estudiantes, docentes, horarios, niveles, asistencias, matrículas y facturación.

Con respecto a los requerimientos planteados anteriormente se ha desarrollo un diagrama general de las funcionalidades las cuales corresponde a las necesidades planteadas por los actores, como se muestra en la figura2.

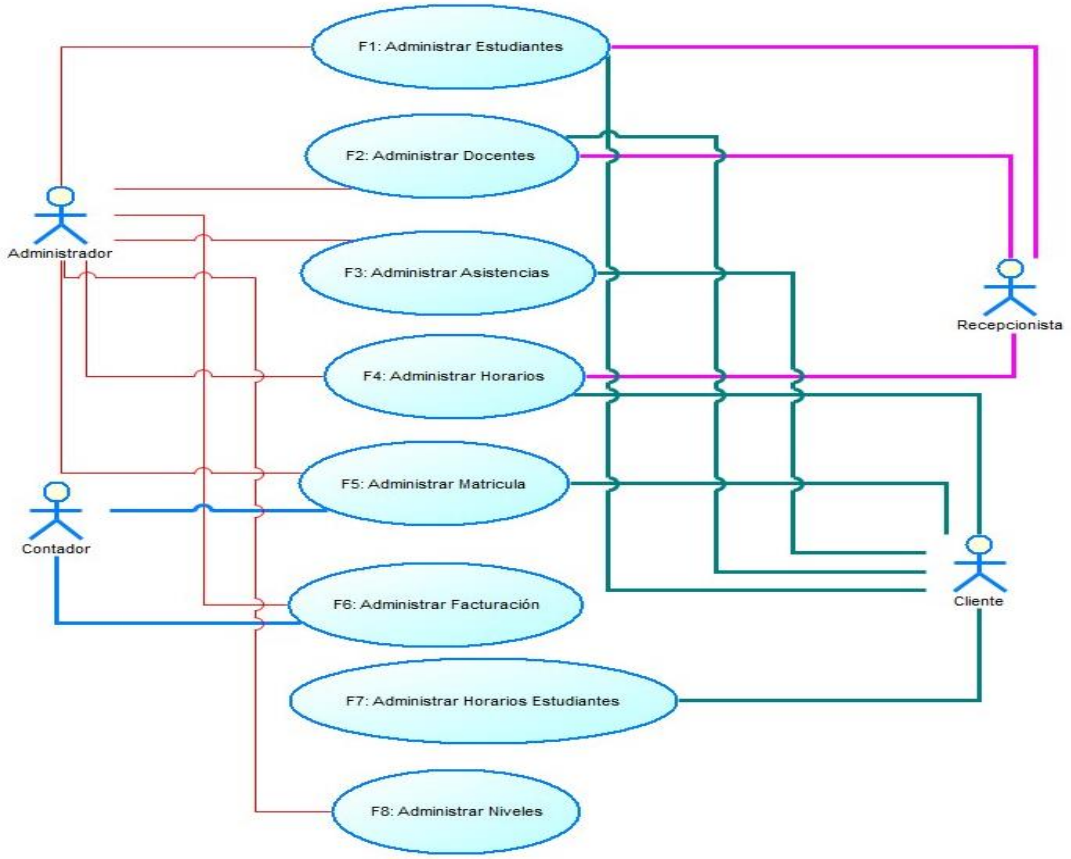


Figura 2. Diagrama general de las funcionalidades del sistema administrativo obtenida del levantamiento de requerimientos.
Fuente: Living English Institute

Al mismo tiempo se realizará el detalle individual de cada requerimiento funcional para proceder a implementar sus debidos casos de uso.

4.1. Detalle funcional en la gestión de estudiantes

Debido a la falta de gestión y perdida de información de los estudiantes se pretende brindar una solución óptima para el registro de estudiantes, así mismo verificar la existencia del estudiante, editar o actualizar los datos del estudiante o su representante, además permite su habilitación o des habilitación en caso de que el usuario se retire, dado que no existe un proceso de eliminación sino un borrado lógico, véase la figura 3.

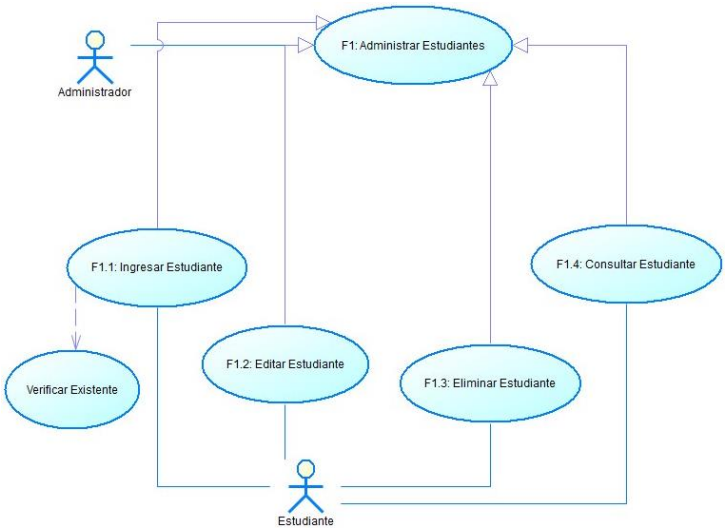


Figura 3. Diagrama funcional de la administración de estudiantes con sus debidos procesos de: ingreso, edición, búsqueda y eliminación lógica.
Fuente: Living English Institute

4.2. Detalle funcional en la gestión de docentes

Por lo que se refiere a los docentes, el instituto tiene la apertura de contratación de nuevo personal docente, sin embargo no existe un respaldo histórico de los docentes que formaron parte de la institución por lo que se necesita implementar un módulo para el registro de docentes, por lo tanto al ingresar un nuevo docente se debe solicitar la información necesaria para mantener un contacto directo en caso de necesitar de sus servicios, este módulo comprende el registro de docente, la actualización de información en caso de ser necesario y la desvinculación del docente con la empresa véase la figura 4.

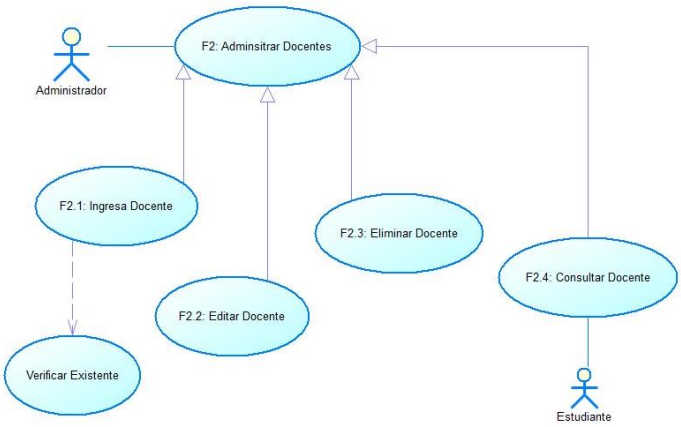


Figura 4. Diagrama especifico funcional de la administración de docentes con sus respectivas acciones: ingresas, editar, eliminar y realizar búsquedas.
Fuente: Living English Institute

4.3. Detalle funcional en la gestión de asistencias

La institución presenta deficiencias en la gestión de asistencias, dado que los registros son entregados en hojas y tienden a extraviarse, al implementar un módulo de asistencias se pretende ingresar los registros de manera que exista un control estricto de asistencias y en caso de incidir con inasistencias se proceda a notificar al representante de su falta a clases, en este módulo se pretende ingresar, actualizar las asistencias en caso de justificación, generar una búsqueda de asistencia en caso de requerir un reporte del estudiante, véase la figura 5.

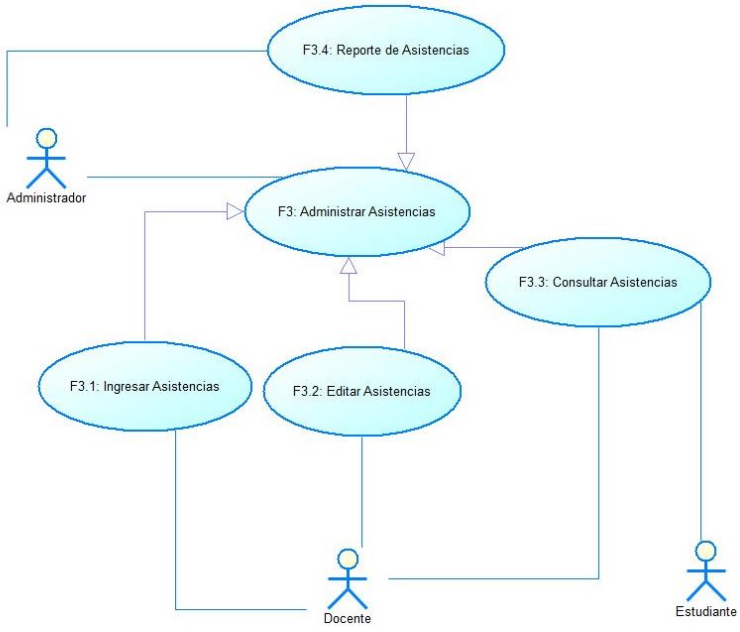


Figura 5. Diagrama funcional de la administración de asistencias, permite agregar asistencias, actualizarlas y realizar búsquedas de un estudiante para emitir un reporte de asistencias.
Fuente: Living English Institute

4.4. Detalle funcional en la gestión de horarios

En la gestión del control de horarios, se requiere administrar varios tipos de horarios, jornadas de trabajo, horarios personalizados en caso de contratación de clases extra, este módulo responde al registro, edición, eliminación lógica y consulta de horarios de acuerdo con la jornada y con el estudiante como se detalla en la figura 6. Un horario puede ser creado de manera personalizada de acuerdo con los días de asistencia del estudiante, además se pueden realizar búsqueda por medio de parámetros que brindan una mayor accesibilidad en los registros, véase la figura 6.

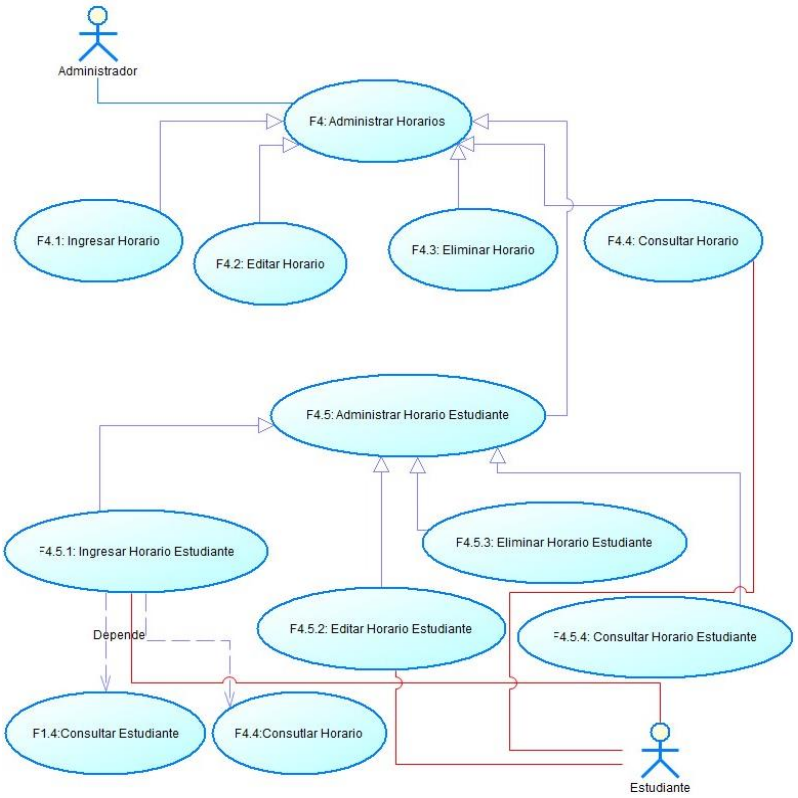


Figura 6. Diagrama funcional de la administración de horarios, en ella se detallan los procesos de creación individual: registro, edición, eliminación lógica y consultas de manera general para cursos anuales, mientras tanto la creación personalizada parte del estudiante que lo solicita.
Fuente: Living English Institute

4.5. Detalle funcional en la gestión de matriculas

En la gestión de matrículas el administrador debe ser capaz de registrar una nueva matricula en base a los estudiantes registrados o un nuevo ingreso, para ellos se ha desarrollado un módulo especifico que sea relacionado con el estudiante registrado o de nuevo ingreso, con lo que se permite registrar una matrícula, editar una matrícula, eliminarla, generar reportes de las matrículas y realizar búsquedas como se detalla en la figura 7. Cabe recalcar que las matrículas están ligadas al proceso de facturación debido que primero el cliente debe matricularse para realizar la facturación de su ingreso. Para ello se ha detallado el siguiente esquema organizacional que brinda una visión del proceso de matriculación.

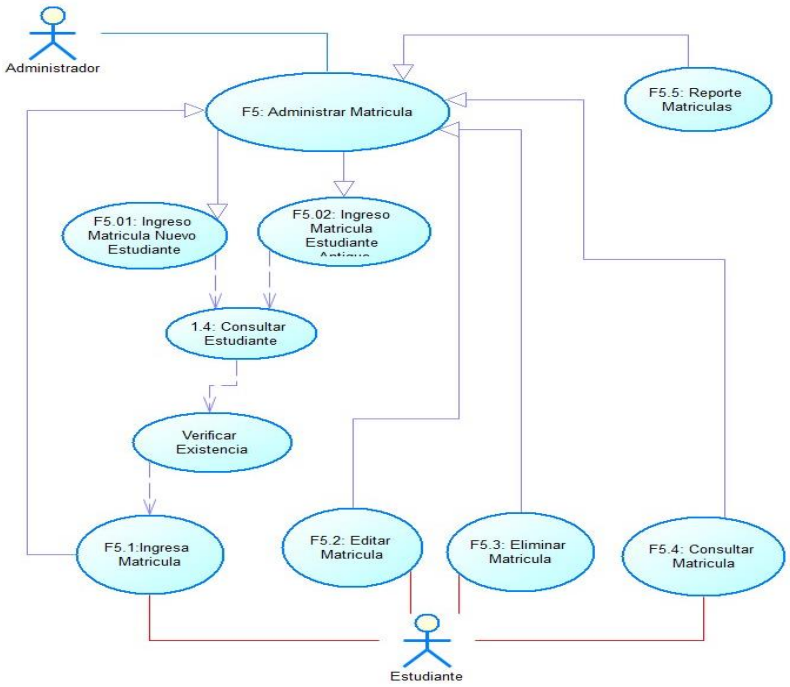


Figura 7. Diagrama funcional de la administración de matrículas, se plantea el registro de matrículas a nuevos estudiantes o estudiantes de reingreso al instituto.
Fuente: Living English Institute

4.6. Detalle funcional en la gestión de facturación

Para la gestión de facturas se brinda una solución óptima ligada a las matriculas o pagos a realizar de las mensualidades, en ella la gestión de facturas debe permitir ingresar una factura en base a la matricula selecciona, en casos internos una factura puede ser editada por error de digitación del encargado de facturar caso contrario las facturas únicamente se podrán anular, se debe implementar un reporte de las facturas emitidas para entregar al usuario y otra para mantenerla como registro histórico de pago, véase la figura 8.

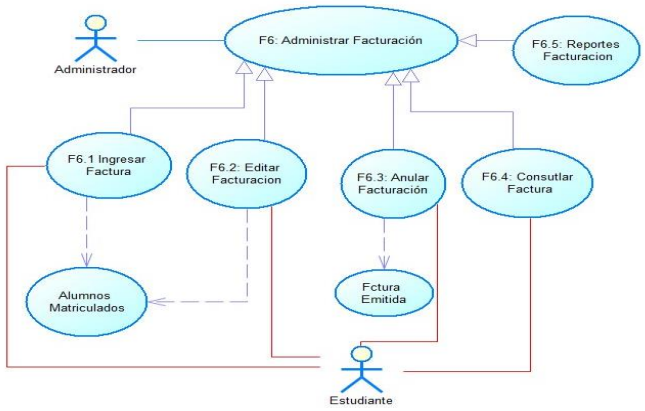


Figura 8. Diagrama funcional de la administración de facturación para el ingreso, anulación y emisión de reportes por cada usuario registrado.
Fuente: Living English Institute

4.7. Detalle funcional en la gestión de niveles

Para gestionar los niveles el sistema debe ser orientado al registro de los cursos a impartirse, debe permitir modificar la duración de los mismos, eliminar o desactivar un nivel y consultar cuales son los cursos impartidos durante un periodo determinado, véase la figura 9.

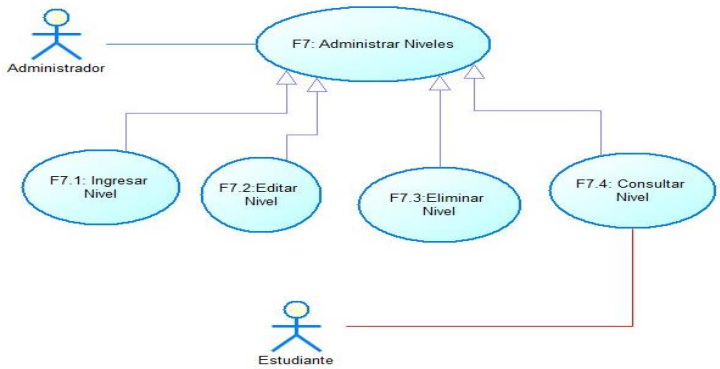


Figura 9. Diagrama funcional de la administración de niveles la funcionalidad debe permitir: registrar, editar, eliminar y consultar por parámetros un nivel.
Fuente: Living English Institute

Luego del análisis de los requerimientos funcionales, se debe detallar los procesos manuales que se realizan en la institución para exponer soluciones automatizadas que generen una mayor productividad y agilicen los procesos administrativos.

4.8. Proceso manual de matrícula

Durante el proceso de matriculación el estudiante se acerca a la institución y solicita matricularse, a su vez el administrador le brinda información de los niveles, los horarios, y el tipo de cursos que presenta la institución. Por otra parte, el usuario selecciona cual es el tipo de curso de interés para inscribirse, luego de haber seleccionado un tipo de programa se procede a tomar la información del estudiante y del representante registrándola en hojas de cálculo sin un control de existencia de estudiantes. Por consiguiente, se le pregunta al usuario si presenta un nivel de conocimiento del idioma o desea empezar desde un curso inicial, si la respuesta es afirmativa se realiza una prueba oral y escrita, luego de realizar la evaluación se indica en qué nivel se encuentra, cuales son los horarios del nivel y los días de asistencia, véase la figura 10.

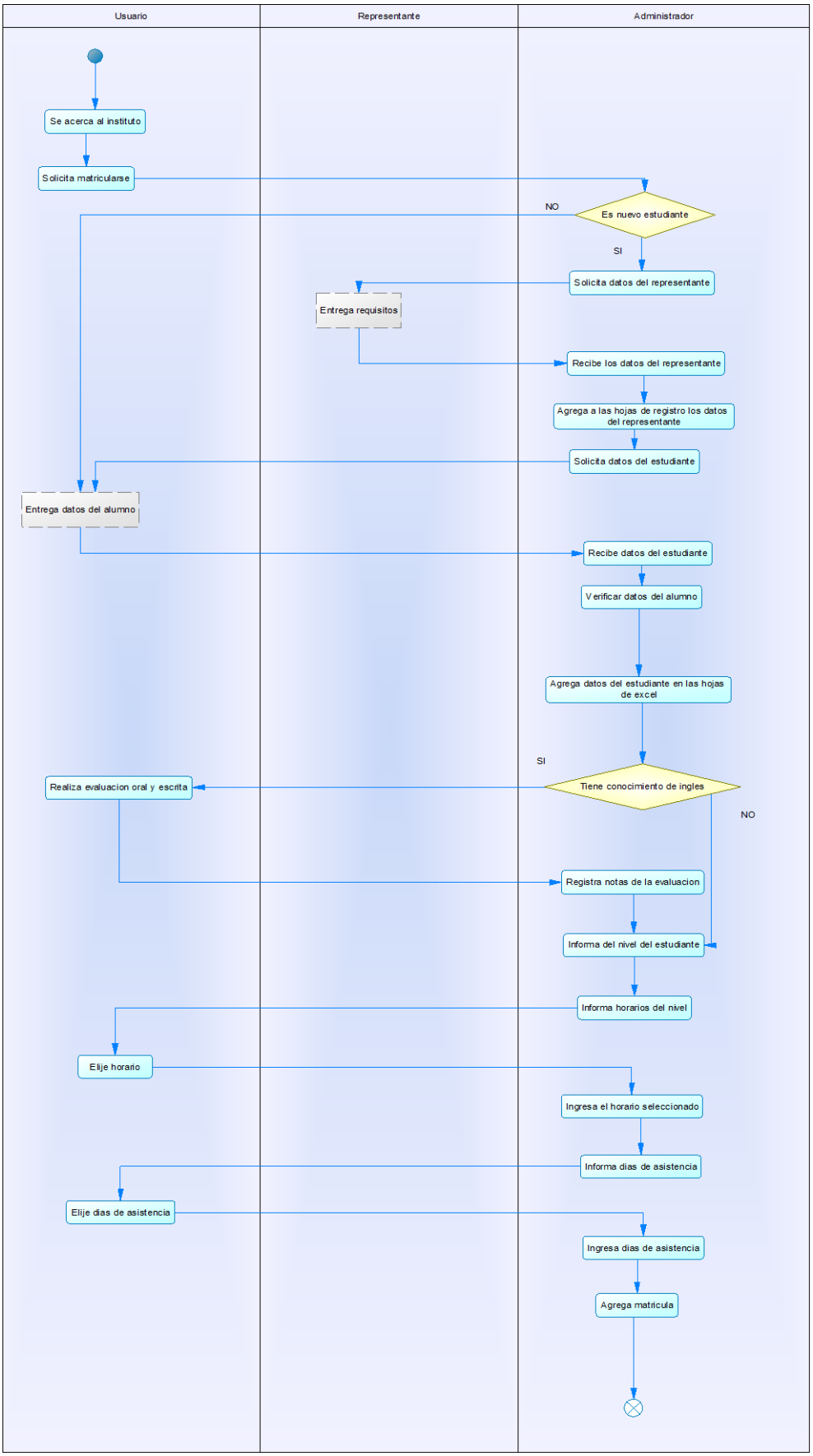


Figura 10. Detalle del proceso manual de matriculación, en el registro de un estudiante.
Fuente: Living English Institute

4.9. Proceso manual de horarios

Actualmente el instituto presenta dos jornadas en un horario de 15:00-17:00 y de 17:30 a 19:30, también existen horarios personalizados para estudiantes que deseen matricularse en un tipo de programa diferente, los tipos de programa son: clases personalizadas, preparación para exámenes, certificaciones y traducciones para determinar los horarios el administrador analiza cuantos estudiantes están matriculados en cada programa académico, si excede la cantidad permitida por cada nivel se creara un nuevo paralelo para no afectar el desarrollo de las clases en un horario ya definido, véase la figura 11 donde se describe la secuencia de creación de los horarios.

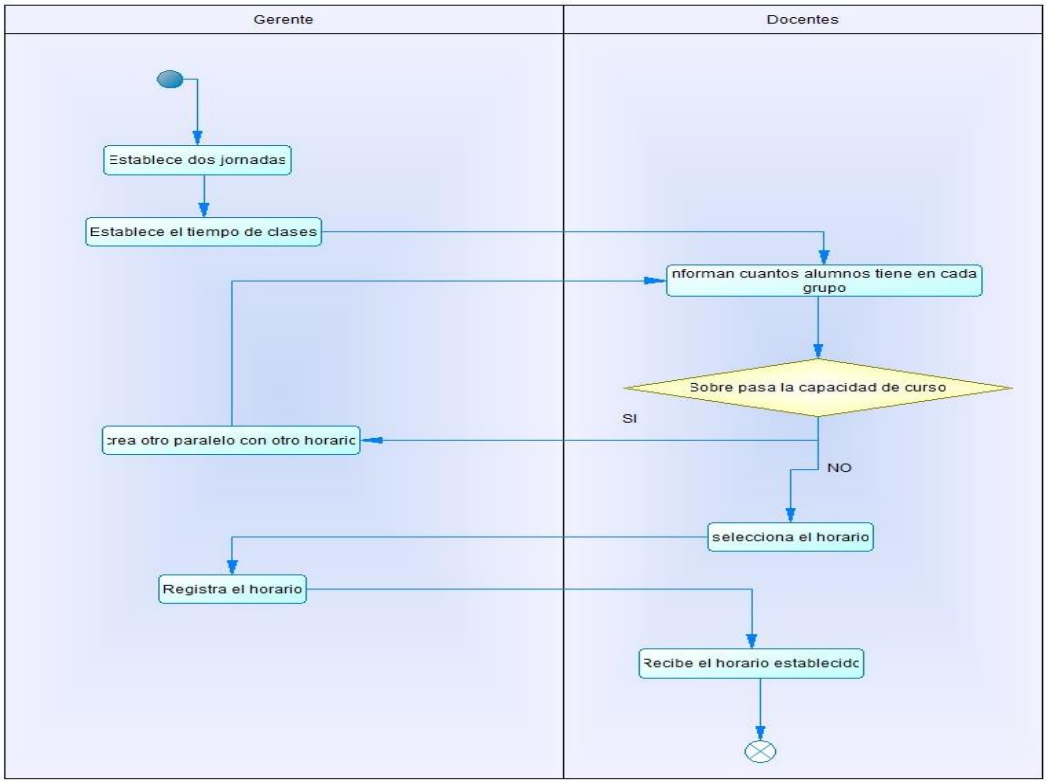


Figura 11. Proceso manual para el registro de horarios, dependiendo del tipo de programa seleccionado.
Fuente: Living English Institute

4.10. Proceso manual de Selección de Horarios del Estudiante

Los estudiantes que se matricularon deben seleccionar la modalidad de sus horarios, lo que se determina de acuerdo con sus facilidades de tiempo, con lo cual el instituto les da la facilidad de asistir a un horario flexible para recibir sus clases, véase la figura 12.

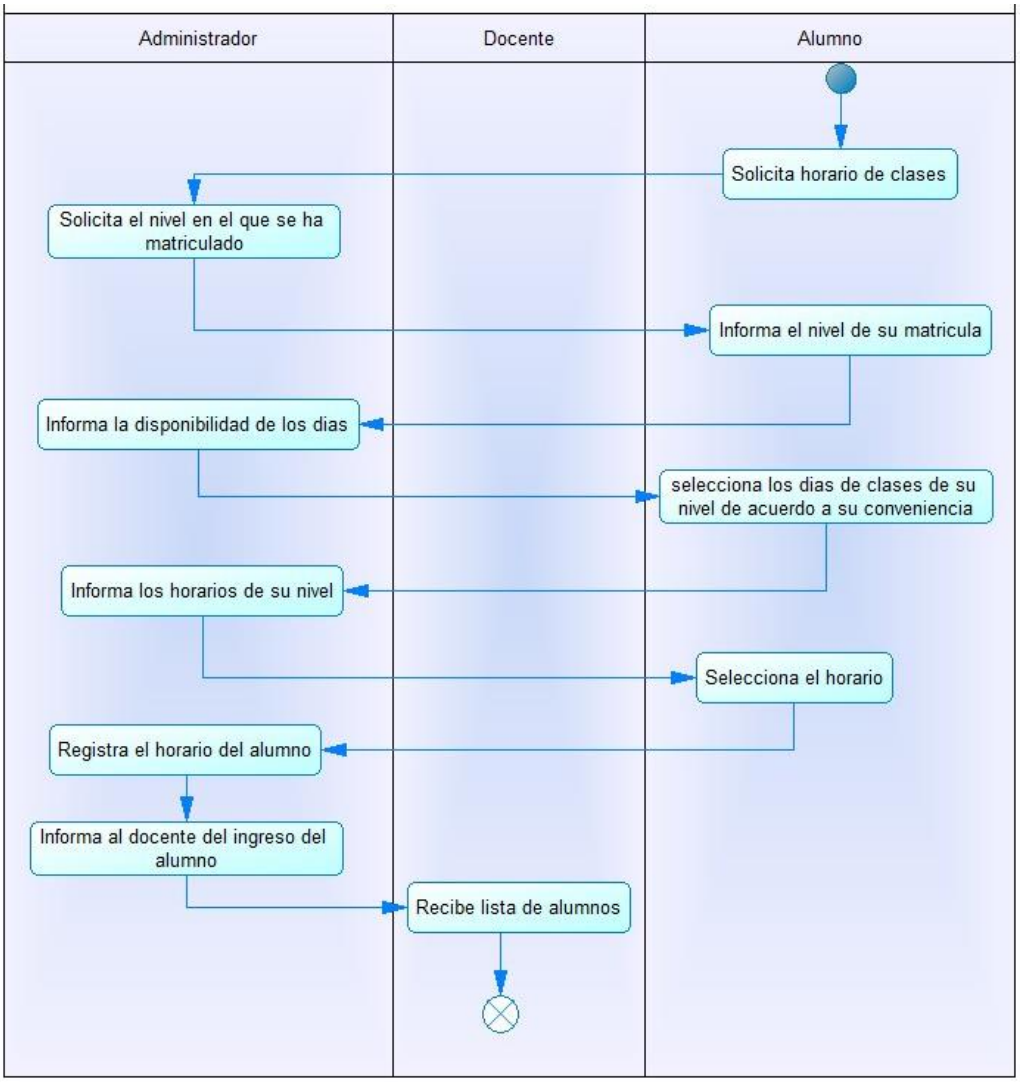


Figura 12. Proceso manual de selección horario por el estudiante.
Fuente: Living English Institute

4.11. Proceso manual de facturación

Para realizar el proceso de facturación, la empresa revisará si el estudiante está matriculado. Así, se procede a rellenar los datos del estudiante a un comprobante físico el cual será entregado al estudiante. Seguido a esto, se le pregunta la forma de pago, la cual puede ser en efectivo, por transferencia bancaria o cheque. Cuando el estudiante ejecuta un previo pago se le remite un comprobante de pago y se le retiene la factura hasta cumplirse el pago en totalidad, véase la figura 13.

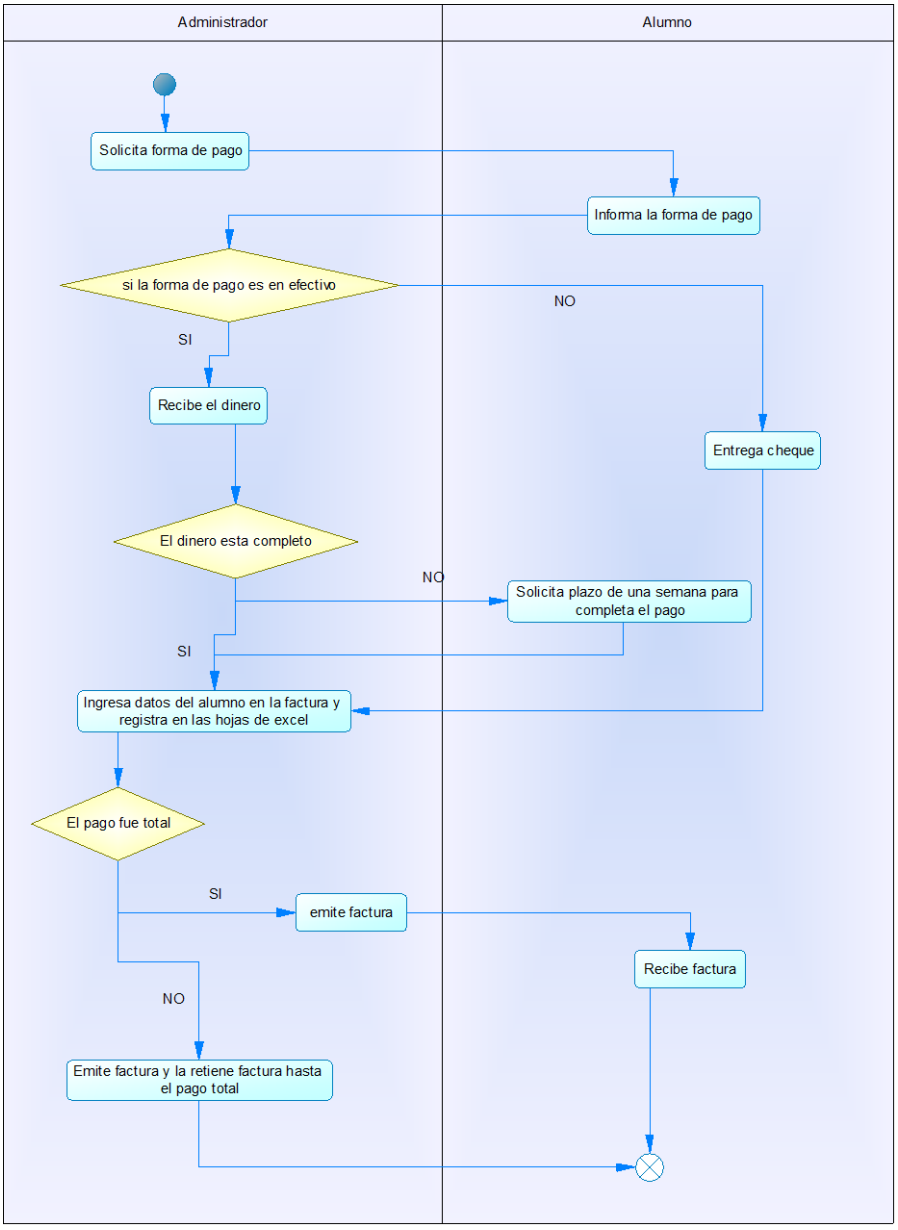


Figura 13. Proceso manual de facturación dentro del instituto.
Fuente: Living English Institute

5. DESARROLLO DE LAS FUNCIONES DEL SISTEMA

Para el desarrollo de las funciones automatizadas del sistema se debe partir de los procesos manuales que fueron detallados en el capítulo anterior. Además, se realizará los casos de uso del sistema para construir la arquitectura base del producto final, cabe recalcar que cada proceso debe tener la aprobación del director para continuar con su implementación.

5.1. Proceso de matrícula

El proceso de matrícula manual tomaba mucho tiempo, debido que los estudiantes no tenían información sobre la empresa. Por lo que era necesario una charla de inducción. Para resolver esto se ha desarrollado un proceso el cual facilita y automatiza esta labor.

Del proceso se debe destacar lo siguiente, la facilidad de usar una computador permite que la matricula no sea un proceso repetitivo de solicitud de datos, dado que la información ya se encuentra dentro del sistema. Así cuando un usuario desea matricularse únicamente se lo debe buscar en la base de datos. Caso contrario se ingresará como un nuevo estudiante y se procederá como lo detalla la siguiente figura, véase la figura 14

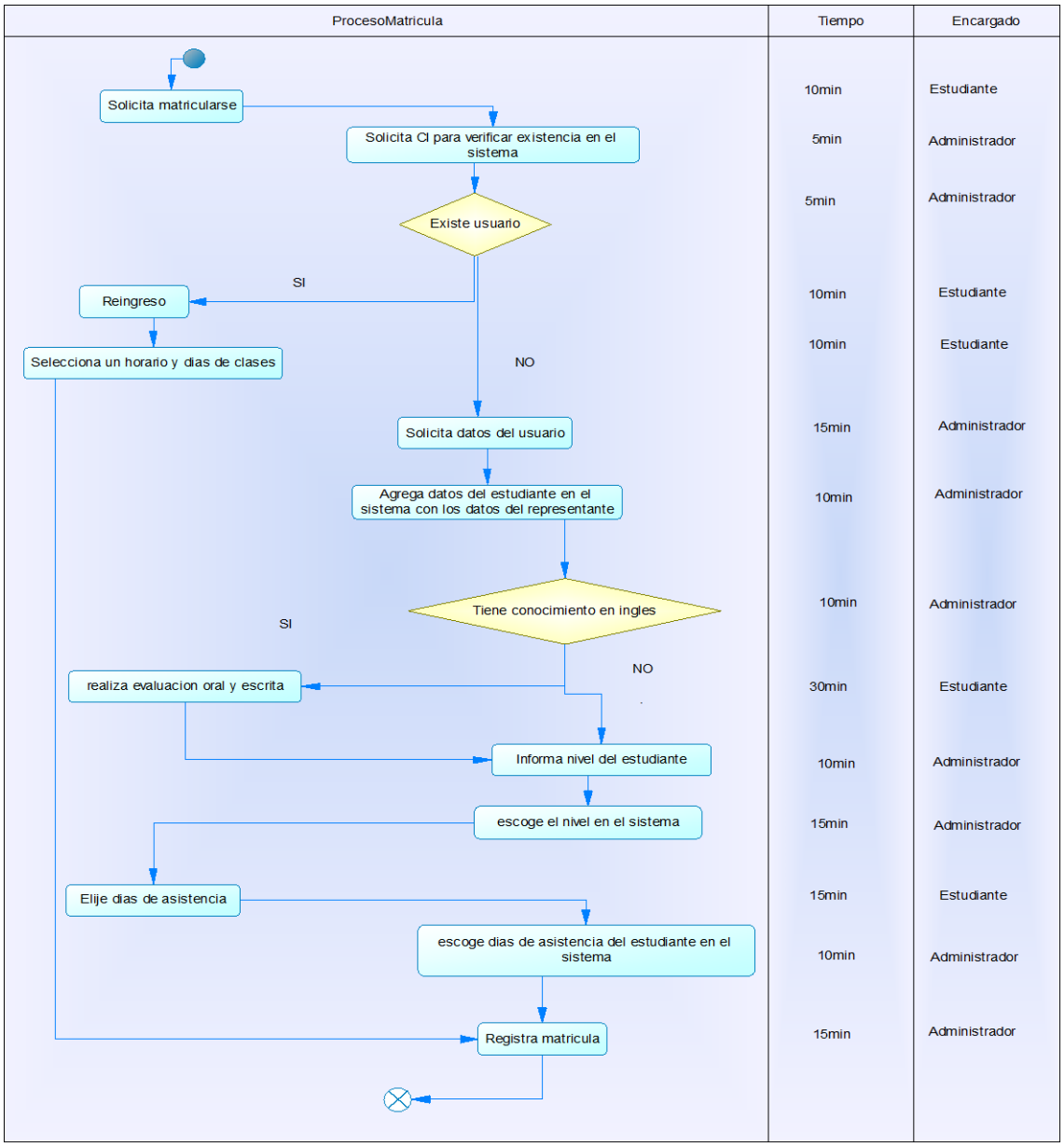


Figura 14. Proceso automatizado de matrícula.
Fuente: Living English Institute

5.2. Proceso de definición de horarios

Dentro de este proceso se busca que los horarios se definan según el cupo de estudiantes que se matriculen en el instituto. Así, se crea el número necesario de paralelos para responder a la demanda de alumnos. Con esto se puede evitar la cantidad excesiva de personas en una misma aula, lo que dificulta la enseñanza, véase la figura 15.

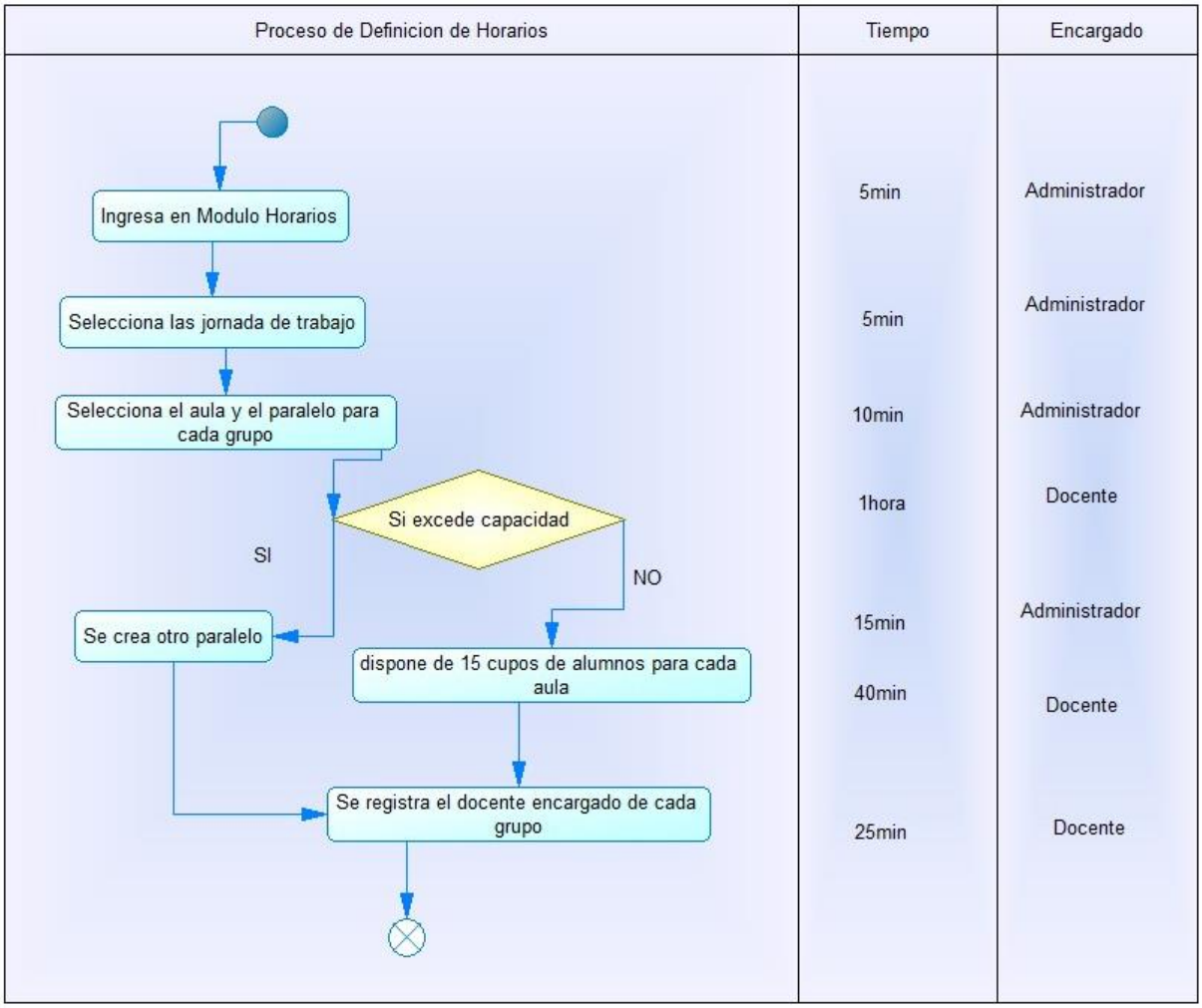


Figura 15. Proceso de definición de horarios, con periodo de tiempos de uso y encargados de área.
Fuente: Living English Institute

5.3. Proceso de selección de horarios del estudiante

Al tener la ventaja de que el instituto es bastante flexible con los estudiantes, se ha podido desarrollar un proceso el cual le da la facilidad al alumno de personalizar su horario. Entre las opciones se destaca que se puede escoger un horario normal preestablecido por el instituto o se lo puede personalizar de acuerdo a las exigencias de tiempo de cada persona, véase la figura 16.

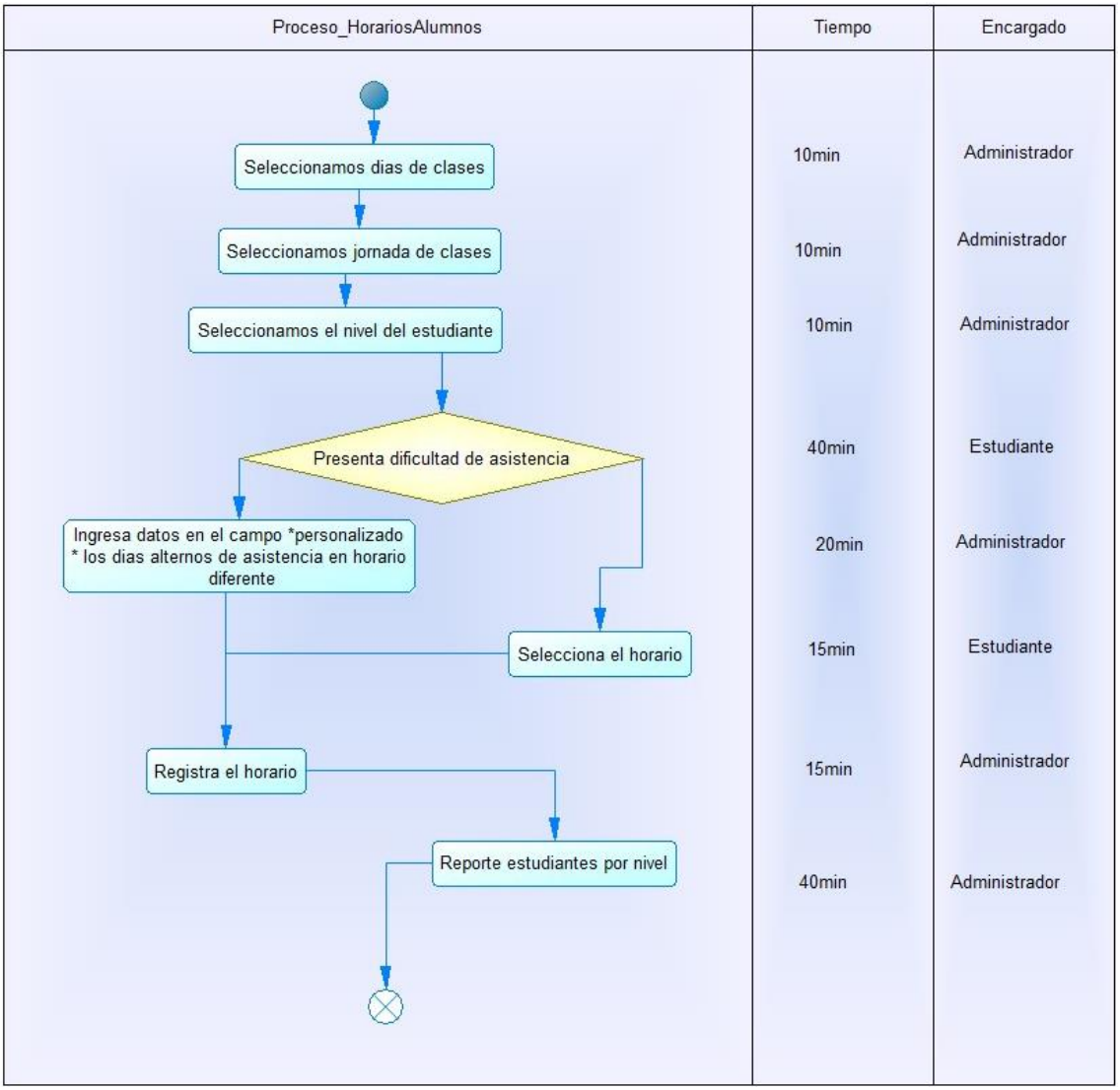


Figura 16. Proceso de selección horario estudiantes, con detalle de tiempo en uso y encargados.
Fuente: Living English Institute

5.4. Proceso de facturación

Este proceso nos permite gestionar todos los pagos realizados por los estudiantes hacia el instituto. De esta manera todos los registros van a quedar guardados dentro de una base de datos, lo que permite tener constancia de los mismos y buscarlos rápidamente en caso de que sea necesario. Aumentando la productividad y eficiencia.

De igual manera este proceso genera las constancias necesarias tanto para el instituto como para el usuario, las cuales se pueden imprimir, véase la figura 17.

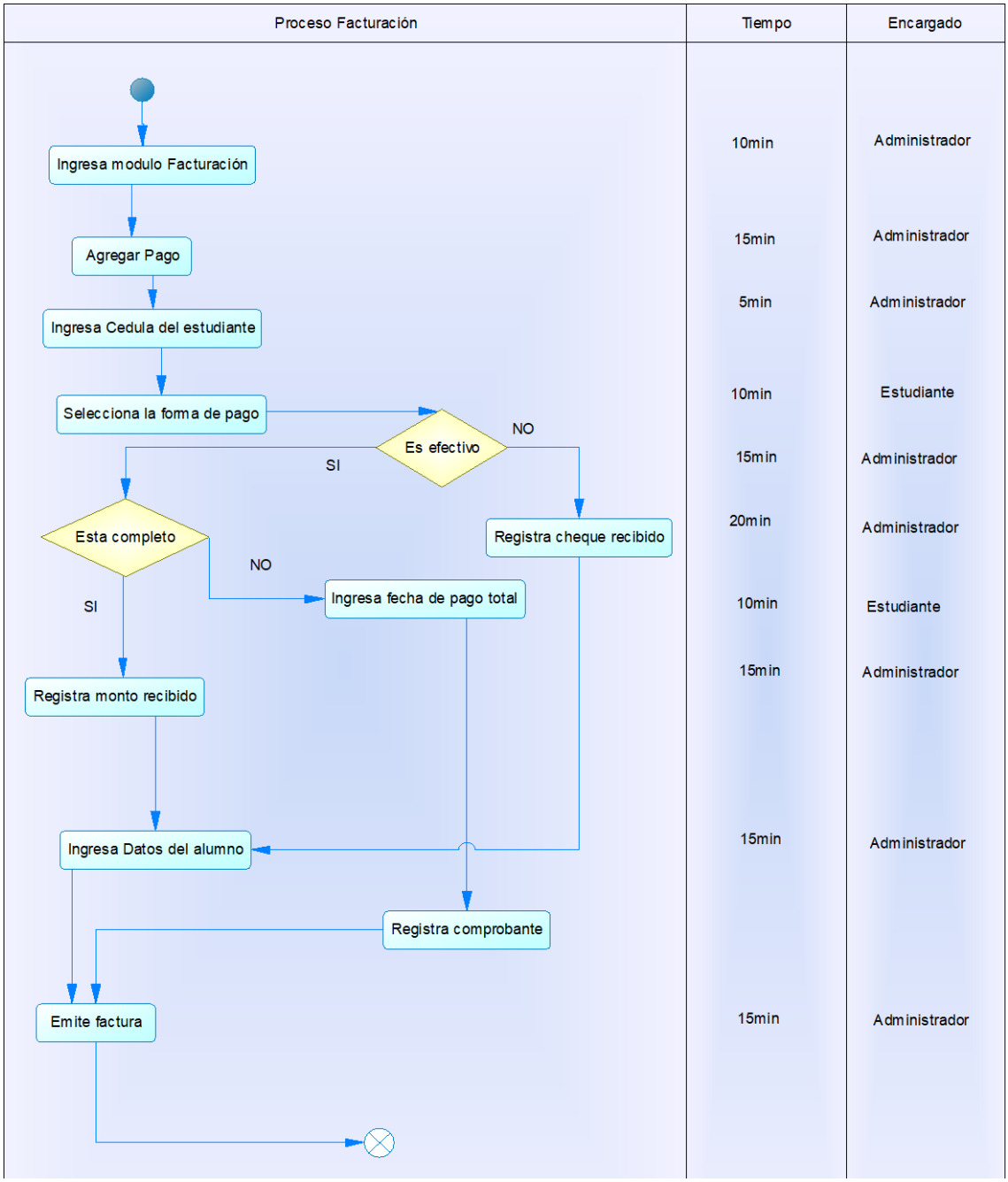


Figura 17. Proceso de control de facturación, detalle de tiempos de uso y personal encargado.
Fuente: Living English Institute

5.5. Proceso de asistencias

El proceso mencionado nos ayuda a llevar un registro de las asistencias de cada uno de los alumnos del instituto, las cuales van a ser correctamente respaldadas en una base de datos para tener constancia de las mismas.

El proceso también emitirá una alarma cuando un estudiante tenga varias inasistencias. Así, el instituto sabrá cuales son las acciones necesarias por tomar, véase la figura 18

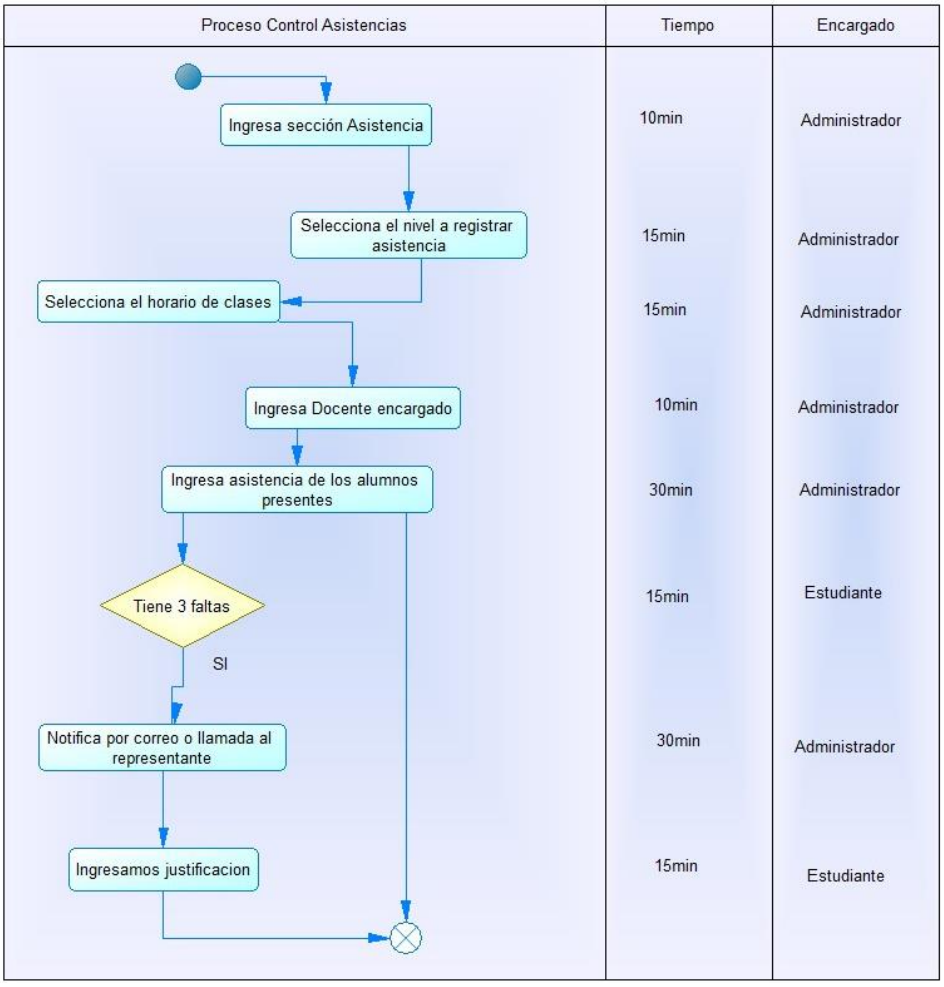


Figura 18. Proceso de control de asistencias, detalle de tiempos de uso y personal encargado.
Fuente: Living English Institute

5.6. DIAGRAMAS DE SECUENCIA

CASO DE USO: Administrar Estudiantes

ESCENARIO 1: Administrador registra datos del estudiante de forma correcta.

Cuando el cliente es nuevo en el instituto y desea registrarse el administrador solicitara todos los datos del estudiante y responsable para poder agregarlos a la base de datos del sistema, véase la figura 19.

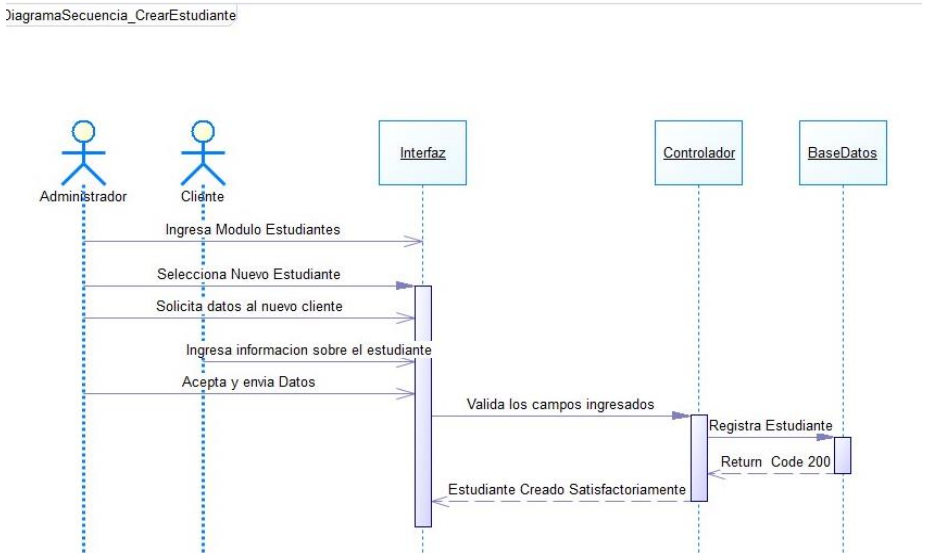


Figura 19. Diagramas de secuencias de estudiantes caso de uso estudiante: secuencia de forma correcta.
Fuente: Living English Institute

CASO DE USO: Administrar Estudiantes

ESCENARIO 2: Administrador registra datos del estudiante de forma incorrecta.

Cuando el cliente es nuevo y el administrador digita datos incorrectos el sistema no ingresara el nuevo registro, dado que hace previa revisión de datos y si existe algún campo incorrecto o vacío se solicitará corregirlo o ingresarlo, véase la figura 20.

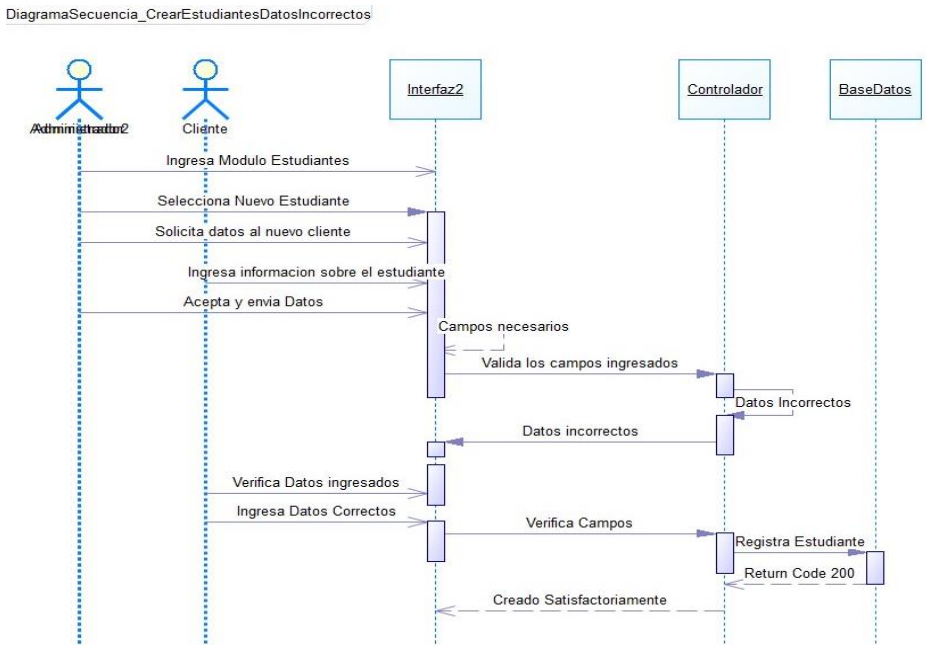


Figura 20. Caso de uso estudiante: se considera incorrecto cuando existe datos incompletos o campos duplicados.
Fuente: Living English Institute

CASO DE USO: Administrar Estudiantes

ESCENARIO 3: Administrador registra datos del estudiante y notifica existencia del estudiante.

El administrador desea agregar un nuevo estudiante, pero dicho estudiante ya consta en la base de datos del instituto, véase la figura 21.

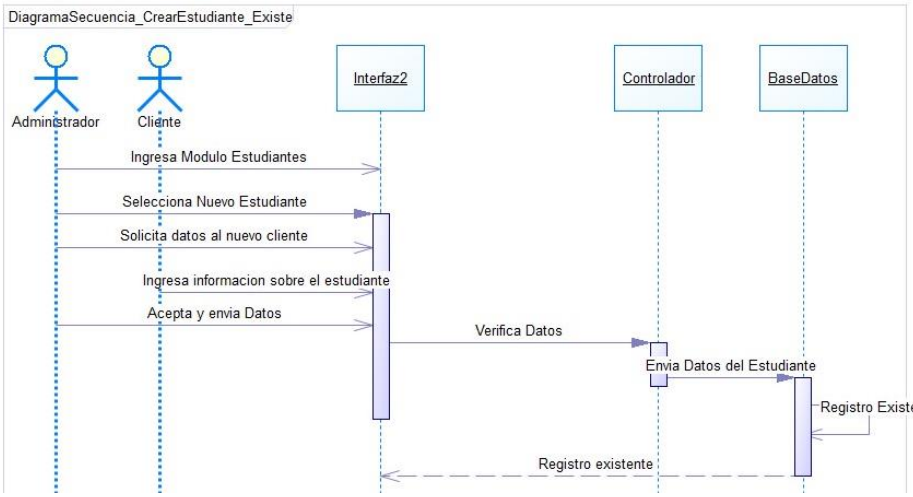


Figura 21. Caso de uso estudiante: datos existentes en la base de datos.
Fuente: Living English Institute

CASO DE USO: Administrar Docentes

ESCENARIO 1: Administrador registra datos del docente de forma correcta.

El administrador desea agregar un nuevo docente calificado para el instituto y este ingresa datos de forma correcta, véase la figura 22.

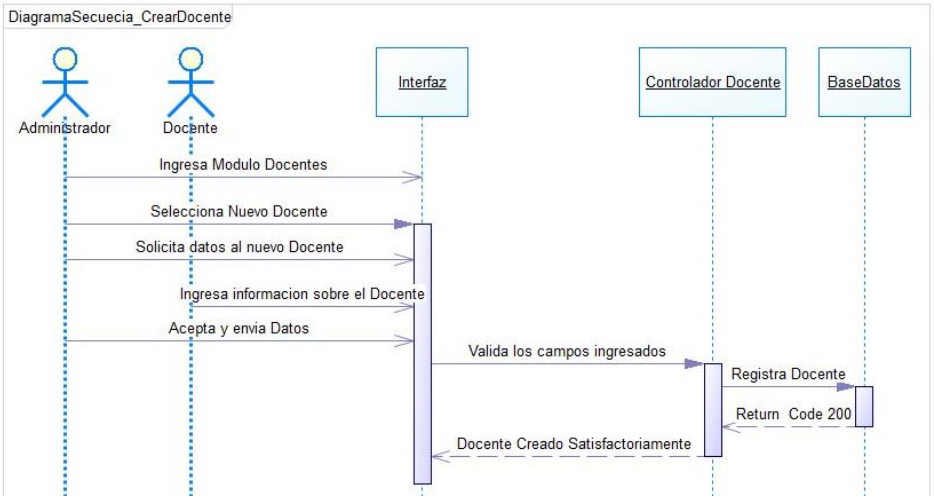


Figura 22. Diagramas de secuencias de docentes caso de uso docente: ingreso correcto de datos.
Fuente: Living English Institute

CASO DE USO: Administrar Docentes

ESCENARIO 2: Administrador registra datos del docente de forma incorrecta.

El administrador desea agregar un nuevo docente en el instituto y este ingresa datos de forma incorrectos o campos vacíos que son necesarios en el registró, véase la figura 23.

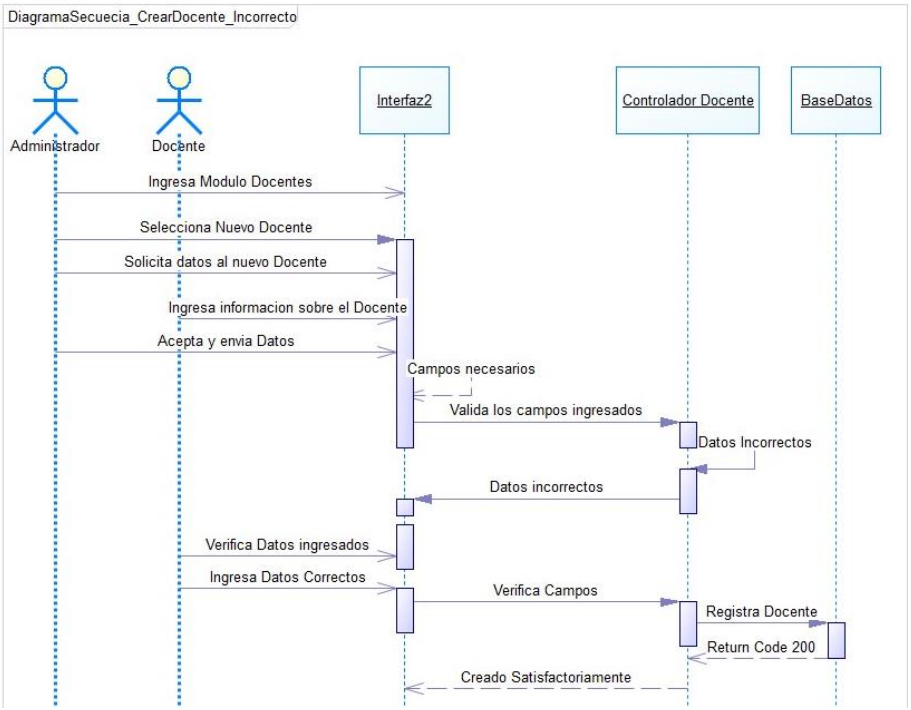


Figura 23. Caso de uso docente: ingreso incorrecto.
Fuente: Living English Institute

CASO DE USO: Administrar Docentes

ESCENARIO 3: Administrador registra datos del docente, pero registro ya existe.

El administrador desea agregar un nuevo docente, pero dicho docente ya consta en la base de datos del instituto, véase figura 24.

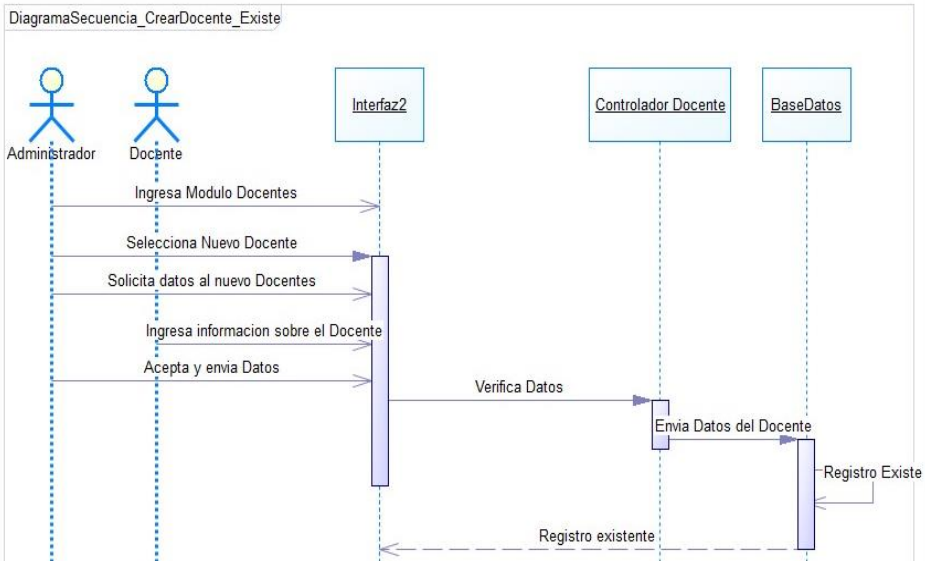


Figura 24. Diagramas de secuencias de docentes caso de uso docente: existente.
Fuente: Living English Institute

CASO DE USO: Administrar Asistencias

ESCENARIO 1: Administrador registra datos de asistencias de forma correcta.

El docente entrega al administrador los datos que se deben agregar de asistencia de los estudiantes de cada nivel impartido, véase la figura 25.

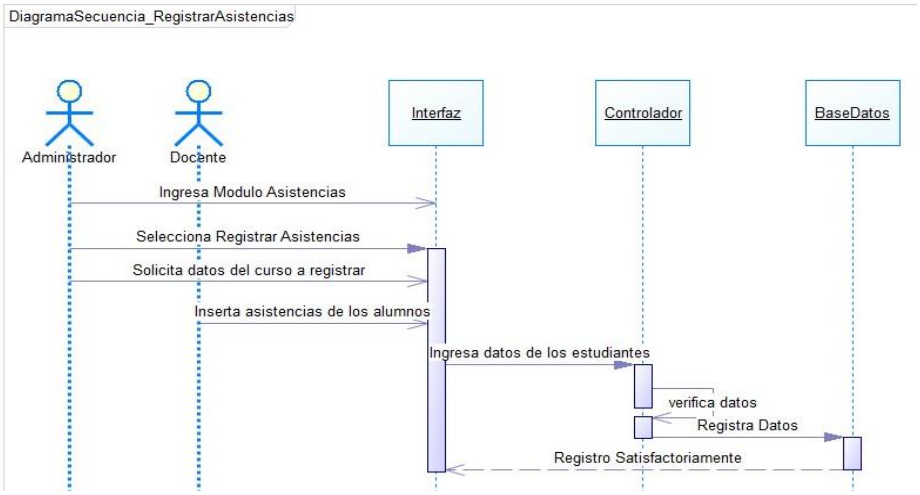


Figura 25. Diagramas de secuencias de asistencias caso de uso asistencia: correcto.
Fuente: Living English Institute

CASO DE USO: Administrar Asistencias

ESCENARIO 2: Administrador registra datos de asistencias de forma incorrecta.

El administrador registra los datos entregados por el docente, pero los datos presentan inconsistencias de datos, véase la figura 26.

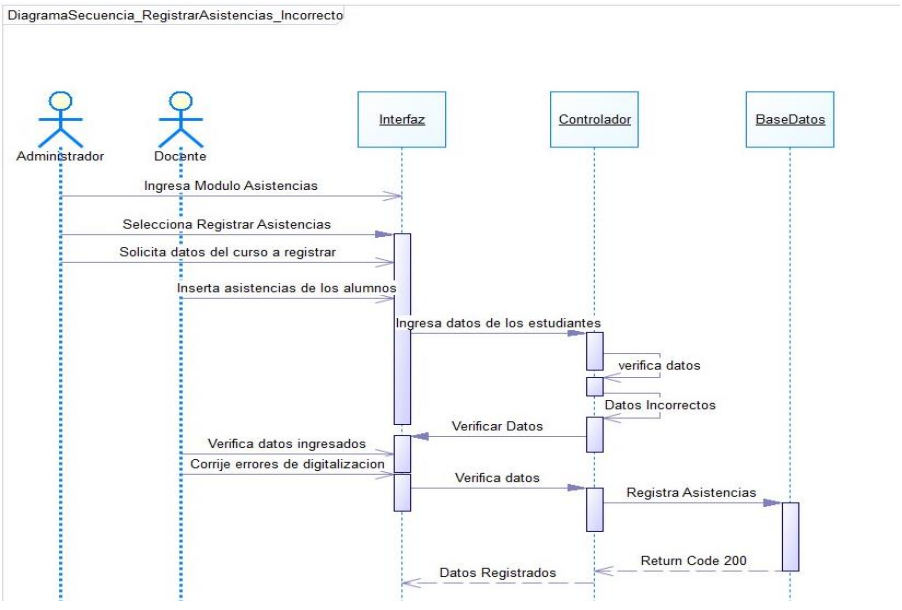


Figura 26. Diagramas de secuencias de asistencias caso de uso asistencia: incorrecto.
Fuente: Living English Institute

CASO DE USO: Administrar Horarios

ESCENARIO 1: Administrador registra datos de horarios de forma correcta.

Administrador agrega nuevo horario, con datos de manera correcta, véase la figura 27.

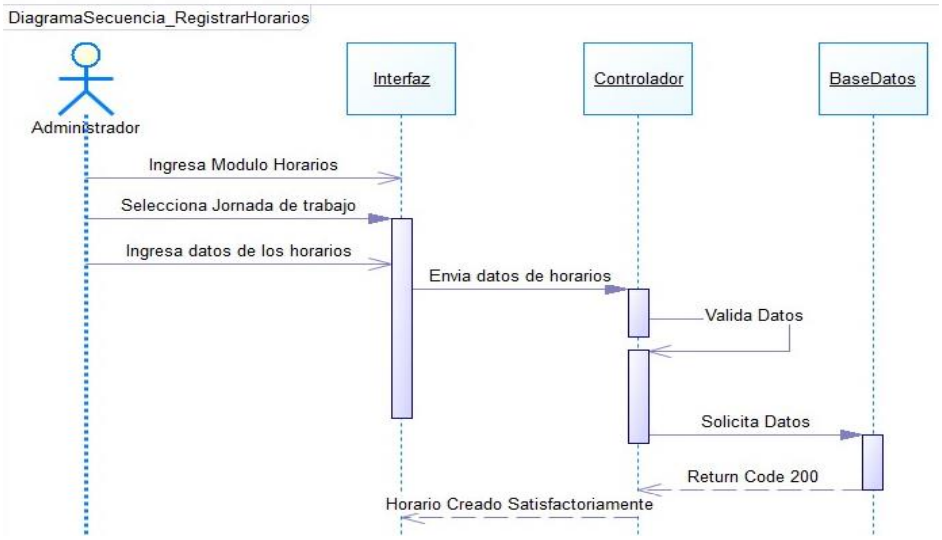


Figura 27. Diagramas de secuencias de horarios caso de uso horarios: correcto.
Fuente: Living English Institute

CASO DE USO: Administrar Horarios

ESCENARIO 2: Administrador registra datos de horarios de forma incorrecta.

El administrador desea crear un nuevo registro, pero el sistema notifica que existen campos vacíos o campos incorrectos, véase la figura 28.

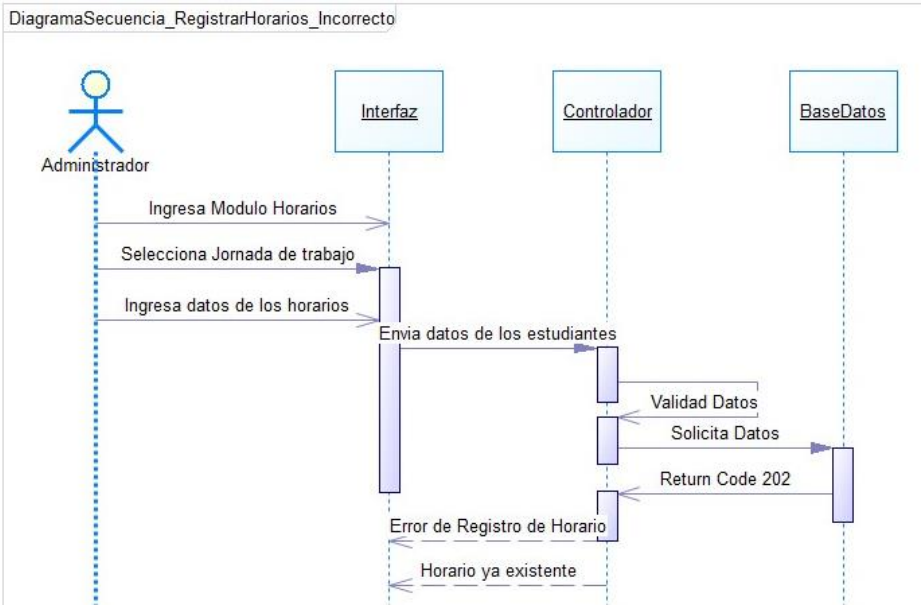


Figura 28. Diagramas de secuencias de horarios caso de uso horarios: incorrecto.
Fuente: Living English Institute

CASO DE USO: Administrar Facturación

ESCENARIO 1: Administrador registra datos de Facturación de forma correcta.

El administrador registra la transacción de facturación con datos y detalles correctos, el controlador valida los datos del formulario, el modelo valida que no se presenten excepciones de datos duplicados, ejecuta la transacción y esta se procesa. El proceso de facturación emite un comprobante donde se detalla los datos, véase la figura 29.

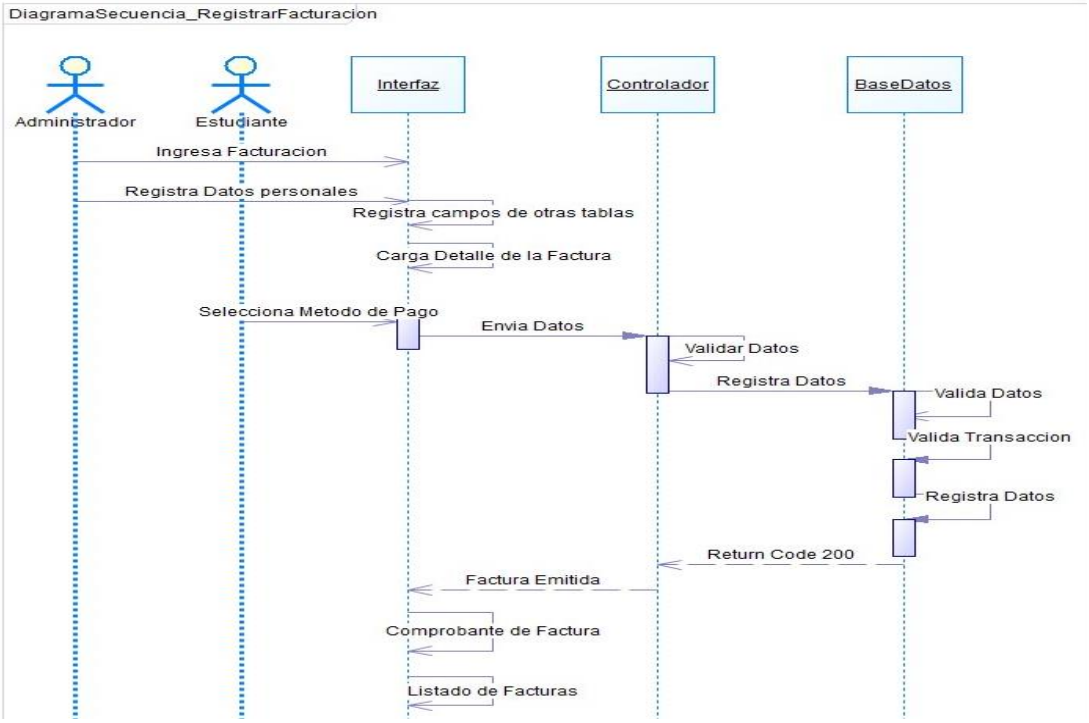


Figura 29. Diagramas de secuencias de facturación caso de uso factura: correcto.
Fuente: Living English Institute

CASO DE USO: Administrar Facturación

ESCENARIO 2: Administrador registra datos de Facturación de forma incorrecta.

Cuando el administrador registra una factura y no han sido registrado los detalles la transacción no se realiza y esto impide que se realice el proceso, véase la figura 30.

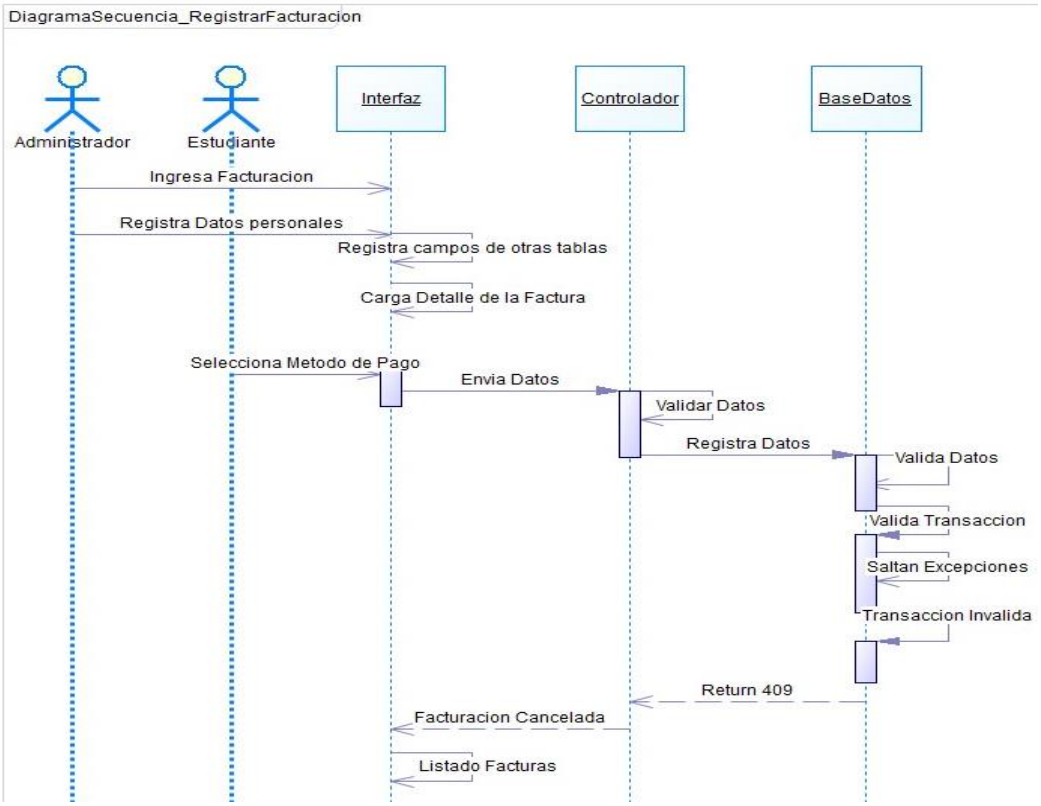


Figura 30. Diagramas de secuencias de facturación caso de uso factura: incorrecto.
Fuente: Living English Institute

CASO DE USO: Administrar Horarios Estudiantes

ESCENARIO 1: Administrador registra datos de horarios Estudiantes de forma correcta. El administrador ingresa al módulo de matrícula y registra el horario en el cual asistirá a clases y continuamente la transacción verifica los datos y registra la matricula, véase la figura 31.

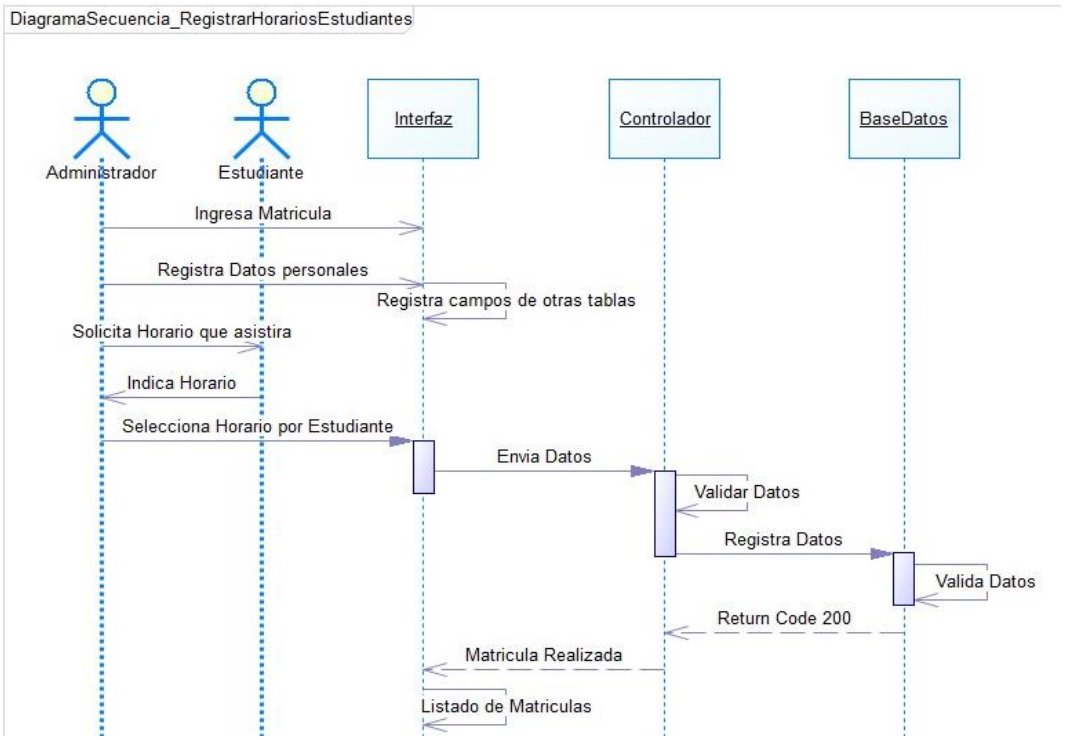


Figura 31. Caso de uso horarios estudiantes: correcto.
Fuente: Living English Institute

CASO DE USO: Administrar Horarios Estudiantes

ESCENARIO 2: Administrador registra datos de horarios Estudiantes de forma incorrecta.

El administrador registra el horario que el estudiante asistirá a clases y si estos datos son correctos conjuntamente a los demás atributos de la matrícula, se procede a verificar y se ejecuta la transacción si un dato es incorrecto o el registro ya existe la transacción falla y anula la matricula, véase figura 32.

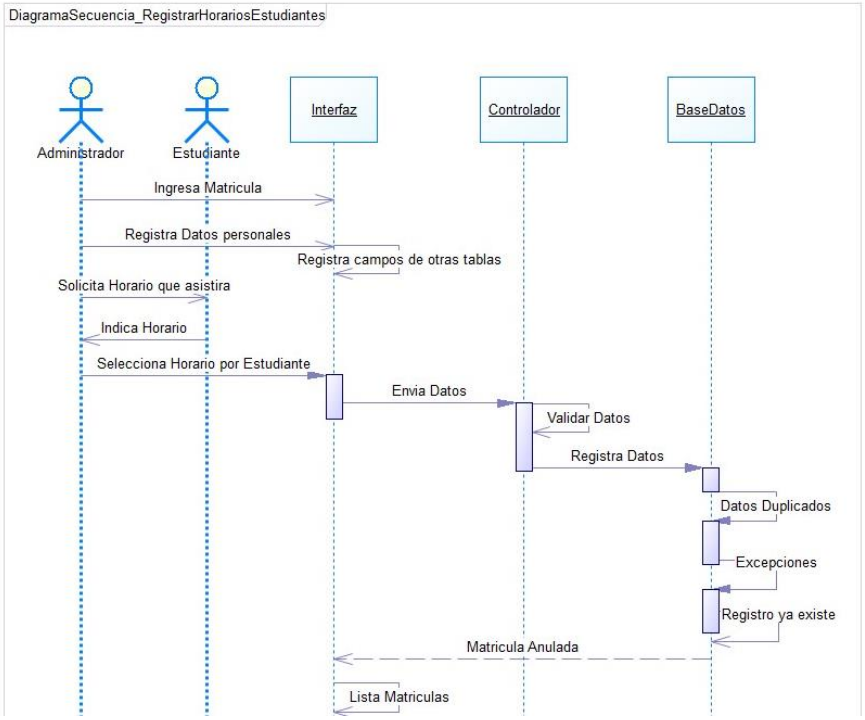


Figura 32. Caso de uso horarios estudiantes: incorrecto.
Fuente: Living English Institute

CASO DE USO: Administrar Niveles

ESCENARIO 1: Administrador registra datos de niveles de forma correcta.

Administrador desea agregar un nivel de aprendizaje y este agrega los datos completos y de forma correcta, véase la figura 33.

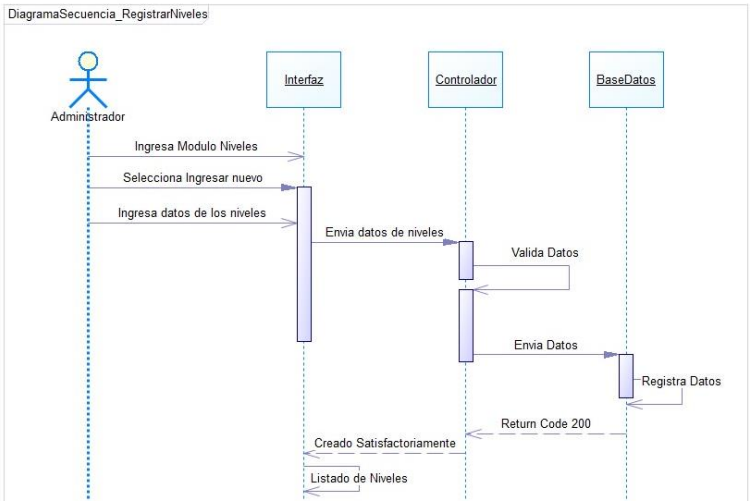


Figura 33. Diagramas de secuencias de niveles caso de uso niveles: correcto.
Fuente: Living English Institute

CASO DE USO: Administrar Niveles

ESCENARIO 2: Administrador registra datos de niveles de forma incorrecta.

Administrador desea agregar un nivel de aprendizaje y los datos son incorrectos o están vacíos, véase la figura 34.

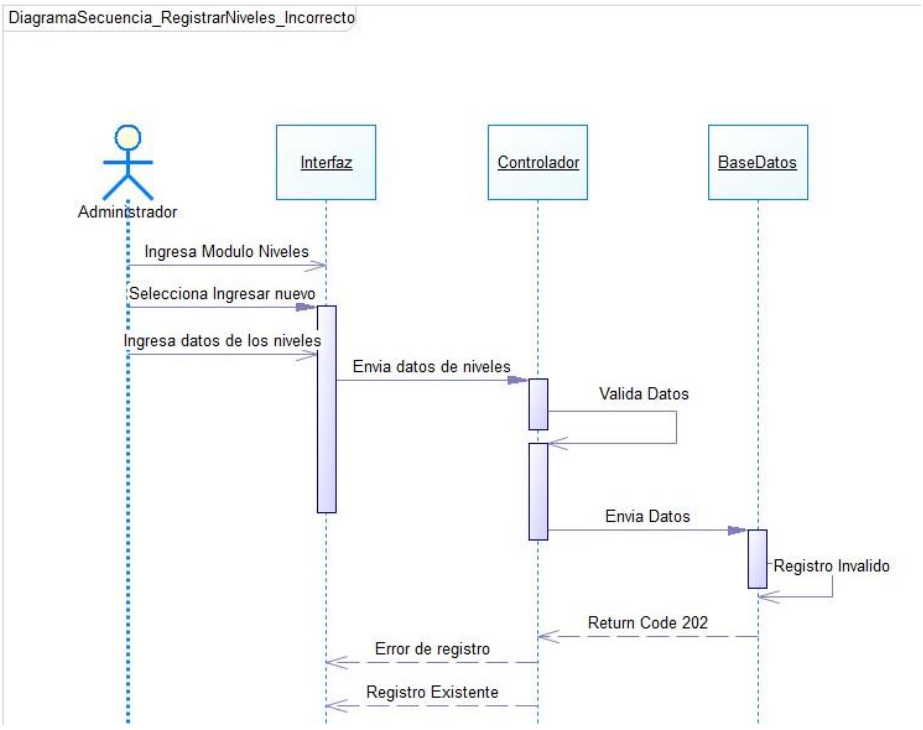


Figura 34. Diagramas de secuencias de niveles caso de uso niveles: datos incorrectos o campos vacíos.
Fuente: Living English Institute

6. IMPLEMENTACIÓN DEL SISTEMA

Durante la etapa de implementación, se seguirá una guía de pasos necesarios desde el análisis de la base de datos, el framework utilizado, la implementación de los requerimientos anteriormente desarrollados, las pruebas de validación y la configuración del sistema en el hosting.

Durante etapas anteriores se realizó el levantamiento de requerimientos y el desarrollo de los procesos necesarios para automatizar la falta de gestión y productividad de la empresa.

Por tal motivo en esta etapa se debe partir del análisis de la base de datos, el sistema gestor de base de datos debe cumplir con los factores como ser multiplataforma, soporte de transaccionalidad, soporte múltiples motores de almacenamiento, alto rendimiento y bajos tiempos de operabilidad.

Para la implementación del sistema de administración del instituto se ha seleccionado el sistema gestor de base de datos de MySQL debido a que cumple con los parámetros de ser un SGBD multiplataforma, soporta procesos transaccionales, es una herramienta open source, tiene una simplicidad y un alto rendimiento a la hora de ejecutar diferentes tipos de consulta y presenta tiempos de respuesta bajos para realizar las tareas necesarias en los procesos desarrollados para el sistema. Además, presenta una herramienta visual para la creación de los diseños de la base de datos, en dicha herramienta se pueden analizar los tiempos de respuesta, la ejecución y administración de consultas SQL.

Cabe recalcar que MySQL ofrece velocidad, flexibilidad y control de roles y usuarios para realizar procesos por medio de permisos asignados a cada usuario. Presenta una adecuada seguridad y estabilidad a la hora de ejecutar proyectos. Además, tiene una fuerte protección de datos, también permite tener un apoyo en el desarrollo integral.

6.1. Modelado de la base de datos

El presente modelo conceptual, permite identificar cuáles son las relaciones de más alto nivel para entender cuál es la lógica del negocio, en ella se observa cuáles son las entidades que intervienen en los diferentes procesos. El modelo conceptual detallado en la figura 35 permite conocer cuáles son las entidades relacionadas Para el proceso del modelo conceptual únicamente se describen las entidades y no se presenta ninguna clave primaria, como puede verse en la figura 35.

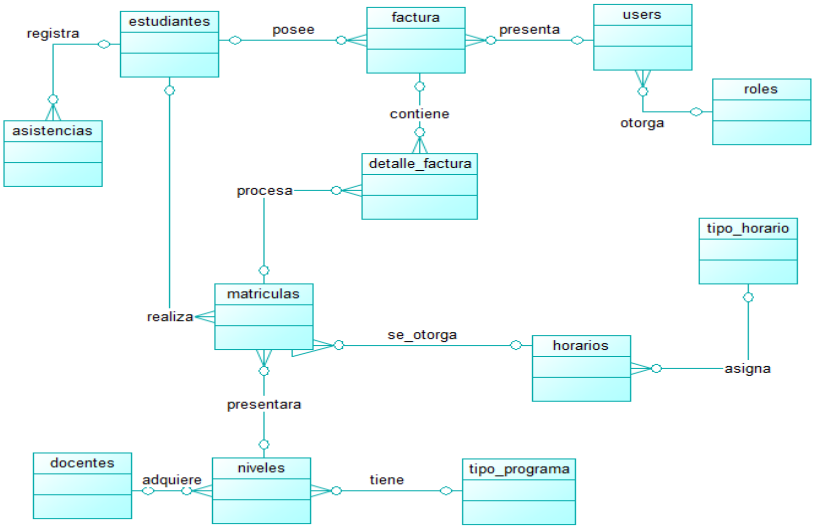


Figura 35. Modelo conceptual de la base de datos: implementación de las entidades sin atributos y claves primarias.

Luego de haber analizado el modelo conceptual, se procede a crear el modelo físico donde se detallará cuáles son los atributos que presenta cada relación y cuáles son las claves primarias, véase la figura 36

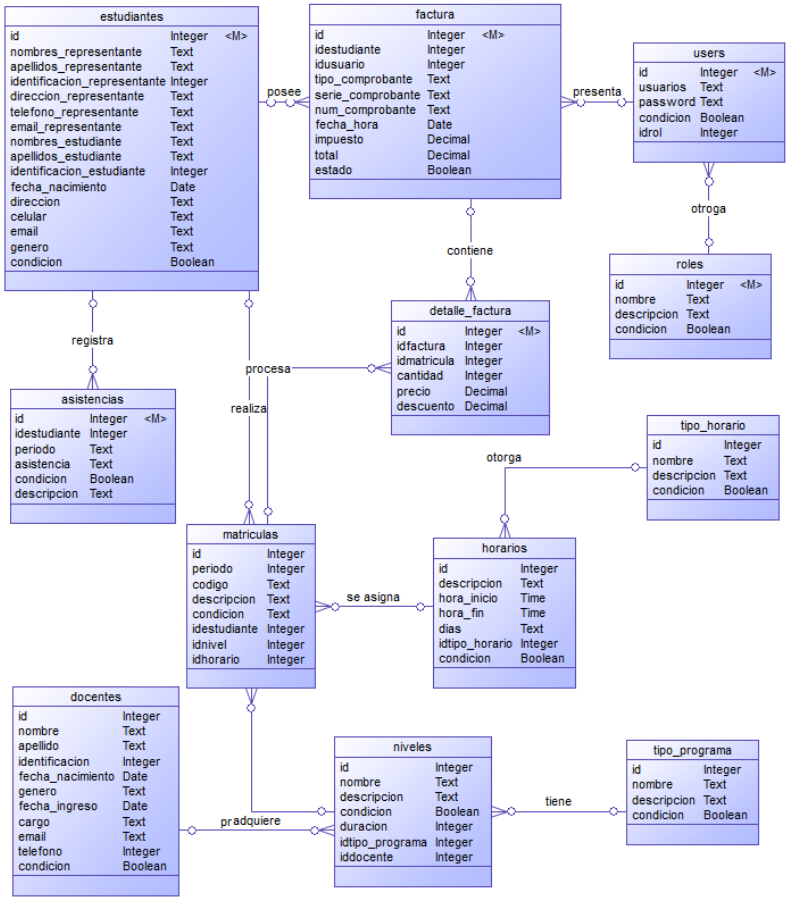


Figura 36. Modelo físico de la base de datos: implementación de atributos y claves primarias

Así mismo, el análisis físico de la base de datos con sus atributos y claves primarias se procede a crear el modelo entidad relación, el cual brinda las relaciones: uno a uno, uno a varios, varios a varios, esta última necesita crear tablas intermedias para garantizar la atomicidad de los datos y la transaccionalidad, véase la figura 37.

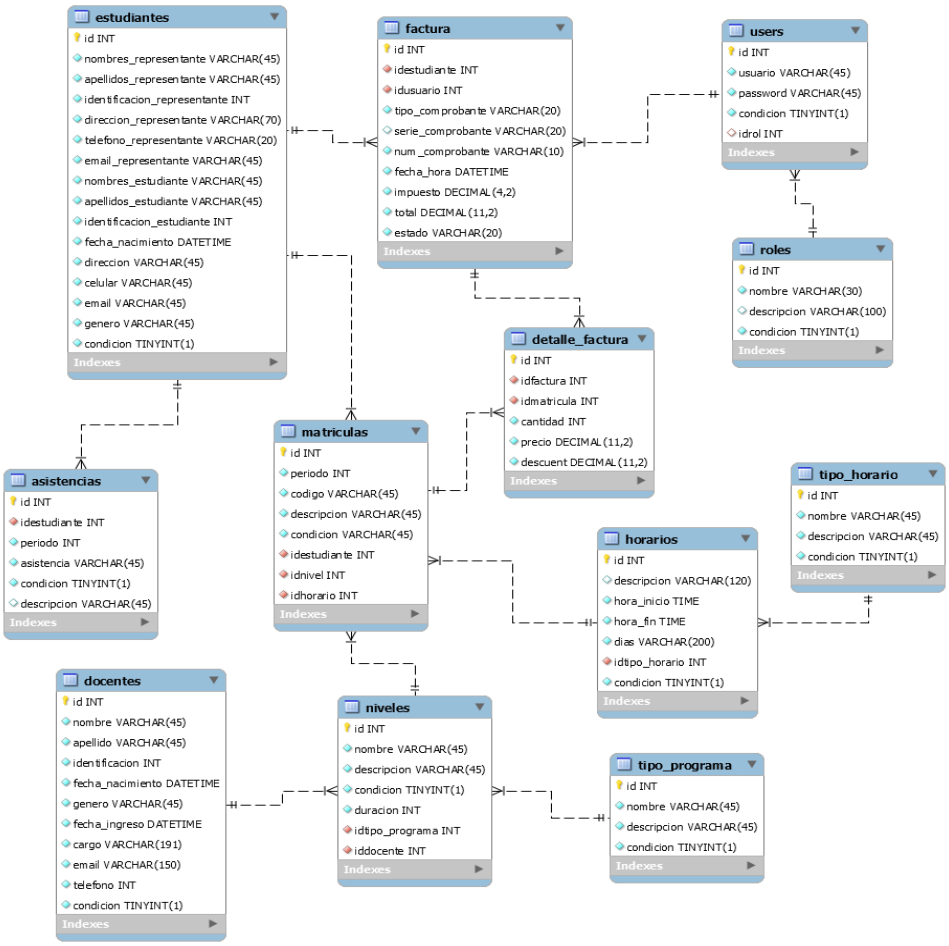


Figura 37. Modelo entidad relación: representada con las relaciones que intervienen en el proceso administrativo y gestión de procesos.

Al haber modelado la base de datos que será integrada se obtiene como resultado el pilar que sustentara la administración de datos.

6.2. Herramientas e integración de procesos en sistema

El segundo aspecto necesario para la implementación del sistema es describir el uso de la herramienta necesaria para crear los procesos que gestionaran la administración del instituto.

Para el desarrollo del sistema se ha seleccionado dos tipos de frameworks que permitan crear un sistema robusto, el cual responderá a las entidades creadas en el modelado de la base de datos. El primer framework seleccionado es LARAVEL un framework basado en el lenguaje php, el cual nos brindara la parte medular del sistema. En Laravel se ha desarrollado la lógica del negocio, toda la parte transaccional que permite controlar las acciones que realizaran los procesos desarrollados. Como segundo framework se ha seleccionado VUEjs, una herramienta que responde al lenguaje de programación Javascript, el uso de VueJS está integrado en el sistema como el soporte visual para crear interfaces que serán utilizadas por cada uno de los usuarios finales. En esta herramienta se crean los módulos y todo lo necesario a la parte visual del sistema. A VueJS se integran los controladores por medio de peticiones HTPP con la librería AXIOS la cual hace de intermediario entre la parte visual y la lógica del negocio que está desarrollo en LARAVEL.

Al integrar el proyecto con estas dos herramientas obtenemos un sistema robusto, seguro y eficiente dado que las cargas de ejecución han sido distribuidas en partes iguales de procesamiento.

A continuación, se detalla cuáles son las principales interfaces creadas y para el desarrollo del sistema.

Inicialmente se mostrará la interface de Login, la cual corresponde al acceso del sistema, donde se solicitará usuario y contraseña para desplegar los módulos internos del software, véase la figura 38.

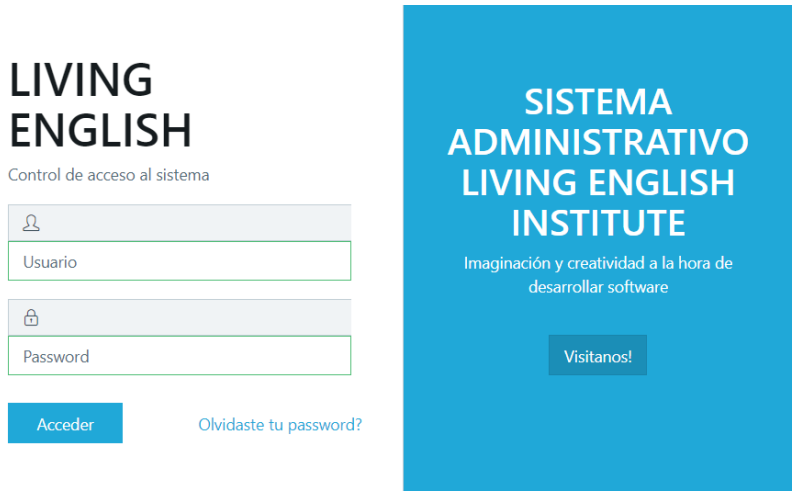


Figura 38. Login de Acceso al sistema: solicita usuario y contraseña para ingresar.

Una vez que el usuario ha ingresado sus credenciales de manera correcta, se desplegará el sistema con sus módulos integrados que corresponden a cada uno de los procesos desarrollados. Dado que el sistema tiene varios actores, la vista que presentará el sistema dependerá del rol asignado, Como se muestra en la figura 39 que desplegará la vista del administrador.



Figura 39. Sistema administrativo: se presenta cada uno de los módulos que responden a los procesos desarrollados.

A continuación, se muestra el módulo de facturación como detalle funcional para demostrar la integración del proceso en el sistema automatizado (véase la figura 40).

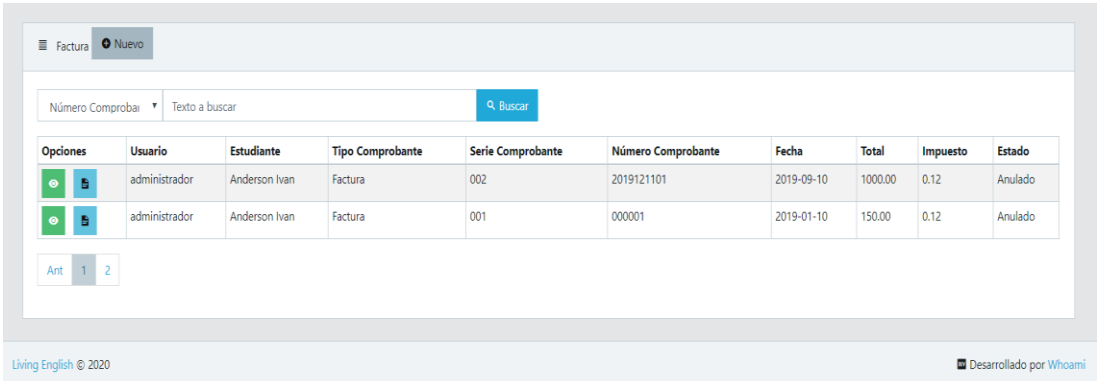


Figura 40. Listado del registro de facturación en el sistema.

Además, se mostrará la interface en la cual se registra una factura o diferente tipo de comprobante (véase la figura 41).

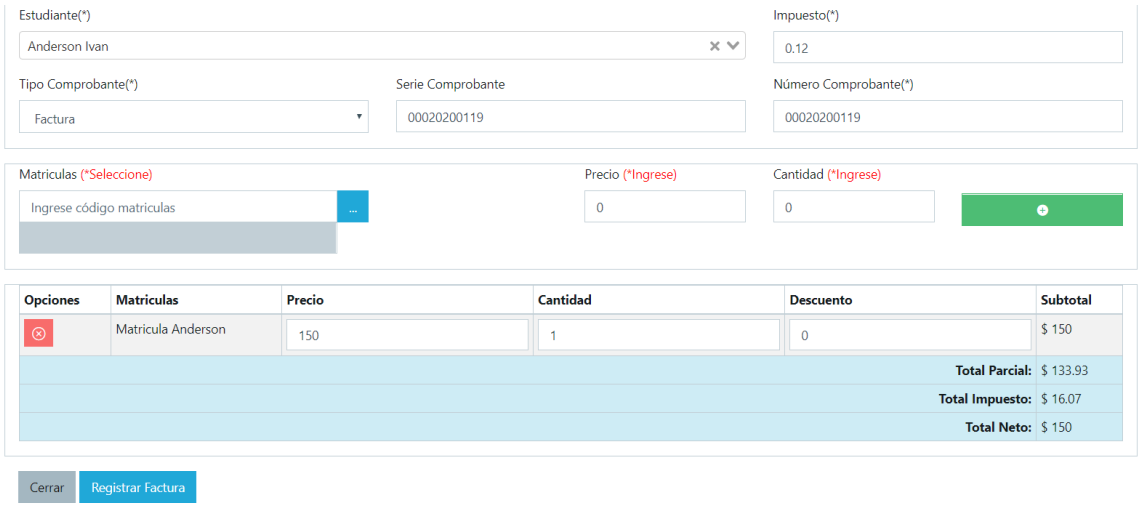


Figura 41. Proceso de facturación: se detalla el nombre del estudiante, el tipo de comprobante, la serie de comprobante, el número de comprobante, la matricula, el precio y la cantidad de matrículas a pagar.

6.3. Pruebas de validación

Para validar que cada componente ha sido creado con satisfacción y que responden a los procesos desarrollados se ha realizado pruebas de validación como se observa en la figura 42.

```
- pagination: {
      total: 12,
      current_page: 1,
      per_page: 10,
      last_page: 2,
      from: 1,
      to: 10
  },
- facturas: {
      current_page: 1,
    - data: [
        - {
              id: 48,
              tipo_comprobante: "Ticket",
              serie_comprobante: null,
              num_comprobante: "00009875",
              fecha_hora: "2019-12-25",
              impuesto: "0.12",
              total: "300.00",
              estado: "Registrado",
              nombres_estudiante: "Mariana",
              identificacion_estudiante: 1720647854,
              usuario: "administrador"
          },
        - {
              id: 47,
              tipo_comprobante: "Ticket",
              serie_comprobante: null,
              num_comprobante: "00009875",
              fecha_hora: "2019-12-25",
              impuesto: "0.12",
              total: "300.00",
              estado: "Registrado",
              nombres_estudiante: "Mariana",
              identificacion_estudiante: 1720647854,
              usuario: "administrador"
          },
```


Figura 42. Pruebas de validación PHP con respuesta del servidor code:200.

Además, se ha realizado pruebas de validación de componentes en los cuales únicamente, detallara cada uno de los atributos registrados de manera correcta, véase la figura 43.

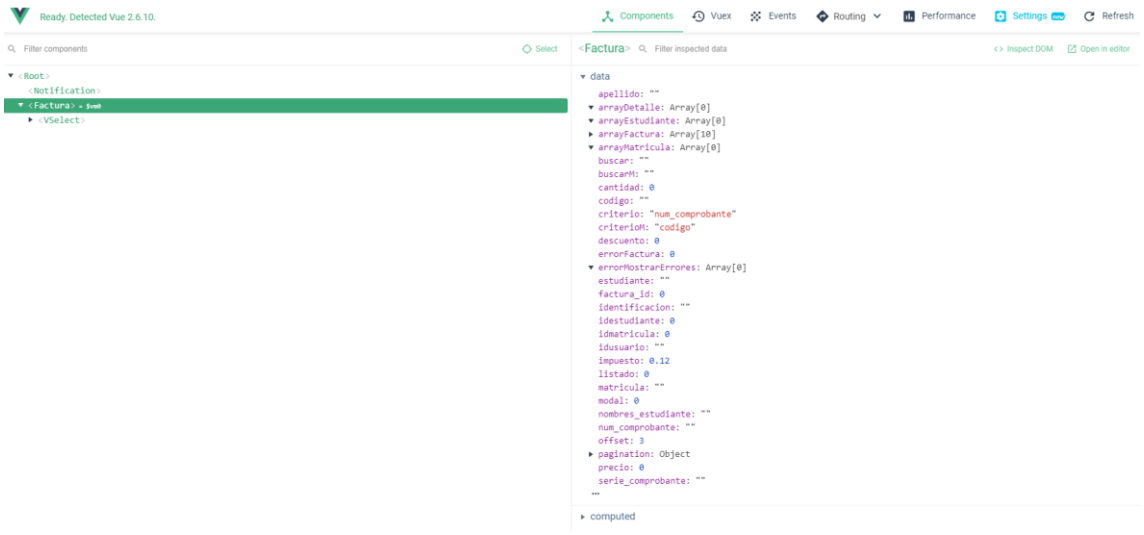


Figura 43. Herramienta de pruebas VUEJS: validación de métodos y atributos procesados de manera correcta.

6.4. Configuración del proyecto en el Hosting

Para configurar el producto en un servidor, se ha contratado un hosting profesional. En el cual se ha hecho las configuraciones adecuadas como son: la configuración de la base de datos, un usuario y una contraseña, se ha configurado el proyecto con las credenciales indicadas por el proveedor del hosting (datos confidenciales).

Luego de contratar el hosting se procede con la compra del dominio y certificados SSL para lograr que el servidor sea de fuentes confiables a la navegación, el nombre del dominio corresponde a https://livingenglishinstitute.ec

Cabe recalcar que los datos son confidenciales los cuales no se permite visualizar debido que presenta usuarios y contraseñas de acceso al sistema. Para demostrar que el producto a sido configurado totalmente en el hosting se procede a mostrar la página oficial del sistema con el dominio especificado anteriormente, véase la figura 44.



Figura 44. Integración del sistema: Dominio de acceso especificado como livingenglishinstitute.ec

6.5. Discusión

Luego del análisis, levantamiento de requerimientos, y recolección de datos tanto de la empresa como de diferentes fuentes de investigación como: libros, informes, manuales y proyectos, se facilitó el desarrollo e integración del sistema con herramientas open source, en las cuales se desarrolló toda la codificación para crear la automatización de procesos del instituto Living English, se debe mencionar que la eficiencia del producto responde al diseño *UX* (experiencia de usuario) y la calidad de interfaces corresponden a *UI* (interface de usuario).

La arquitectura integrada en el sistema corresponde al MVC, donde su codificación está dividida en dos partes: backend y frontend, las cuales representa en orden mencionado a la lógica del negocio y este último al diseño visual del sistema.

El sistema presenta una base de datos, la cual fue modelada en el sistema gestor de base de datos MySQL dado que responde a los factores necesarios para la implementación del proyecto.

Finalmente se desarrolló la metodología RUP, la que brinda una estructura funcional base desde el inicio a fin en el desarrollo de software, por tal motivo dicha metodología facilito el análisis, desarrollo y le implementación del software.

7. CONCLUSIONES

Se logró analizar, desarrollar e implementar un sistema administrativo para el instituto Living English, dado que permite agilizar los procesos manuales llevados en la institución, lo cual brindará una mayor eficiencia y productividad.

Al tomar en cuenta las necesidades del sistema y al hacer un análisis de las diferentes bases de datos existentes se pudo determinar que MySQL es el gestor de base de datos, que mejor responde a las necesidades planteadas. Esto se demuestra al comprobar que el sistema no muestra ningún error dentro de las pruebas que se realizaron.

Se ha logrado desarrollar un sistema el cual nos permite gestionar eficientemente lo relacionado a la parte administrativa del instituto de inglés “Living English Institute”. Mismo sistema que controla todos los datos necesarios para llevar a cabo todos los procesos que el instituto maneja normalmente.

Al implementar el sistema desarrollado en un servidor, se pudo comprobar que el mismo trabaja de manera correcta en cualquier computador con una conexión estable a Internet. Esto permitirá que cualquier usuario del instituto, pueda hacer uso del software y conectarse al mismo con gran facilidad.

8. RECOMENDACIONES

Si el sistema cambia de forma significativa se recomienda volver a realizar un análisis para determinar cuál base de datos sería la más óptima. De ser necesario se debería migrar a un nuevo sistema gestor de base de datos.

Se recomienda al instituto tomar en cuenta todas las consideraciones necesarias para que el sistema funcione de la mejor manera y no se presenten dificultades en ninguno de los procesos. Leer el manual de usuario será una tarea imprescindible para todas aquellas personas que vayan a gestionar el sistema. De esta manera los resultados serán los óptimos.

Al ser un sistema susceptible a cambios y posibles mejoras. Se recomienda respaldar la información ingresada con copias de seguridad que garanticen que no se perderá los datos. De esta manera se segura la escalabilidad del software que se ha desarrollado

BIBLIOGRAFÍA

`emfasi. (1 de Noviembre de 2019). *`emfasi Tus resultados, nuestro objetivo.* Obtenido de https://www.emfasi.com/desarrollo-de-aplicaciones-web

Alvarez, M. A. (02 de Enero de 2014). *desarrolloweb.com*. Obtenido de https://desarrolloweb.com/articulos/que-es-mvc.html

Ángel Pisco Gómez, J. J. (2017). *FUNDAMENTOS SOBRE LA GESTIÓN DE BASE DE DATOS.* Manabi: 3Ciencias.

Arias, M. A. (2013). *Introducción a PHP*. IT Campus Academy.

Castillo, A. A. (2015). *Curso de Programación Web. Javascript, Ajax y jQuery*. IT Campus Academy.

Catalinas, E. Q. (2002). *Sistemas operativos y lenguajes de programación.* España: Editorial Paraninfo.

Center, I. K. (2019). *IBM Knowledge Center*. Obtenido de Programación orientada a objetos: https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/jython/clementine/python_object_oriented.html

Cobo, A. (2008). *Diseño y programación de bases de datos.* Madrid: Editorial Visión Libros.

Corporation, O. (2019). *MySQL*. Obtenido de https://dev.mysql.com/doc/refman/8.0/en/introduction.html

de Pablos, C., López Hermoso, J., Martín Romo, S., & Medina, S. (2004). *Informática y comunicaciones en la empresa.* Madrid: ESIC EDITORIAL.

Fernando Luna, C. P. (2018). PROGRAMACION WEB Full Stack 12 - Sitios multiplataforma con Bootstrap: Desarrollo frontend y backend - Curso visual y práctico. *RedUsers*, 24.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.* Madrid: Pearson Educación, S.A.

jQuery, L. F. (2019). *jQuery write less, do more*. Obtenido de https://jquery.com/

Luna , F., Peña Millahual, C., & Lacono, M. (2017). PROGRAMADOR WEB Full Stack. *USERS*, 24.

Mariscal, A. B. (2015). *UF2405 - Modelo de programación web y bases de datos.* España: Editorial Elearning, S.L.

Orjuela, J. J. (s.f.). *Diseño de una arquitectura web distribuida de alta disponibilidad para sistemas de educación a distancia por medio de Oracle WebLogic Server.* Colombia: Juan Jose Orjuela castillo.

Otwell, T. (2019). *Laravel*. Obtenido de https://laravel.com/

Pavón Puertas, J., & Llarena Borges, E. (2016). *Creación de un sitio web con PHP y MySQL.* Bogota: Ediciones de la U.

Quiñonez-Pico, A. (18 de Marzo de 2017). *¿QUÉ ES EL DISEÑO WEB?* Obtenido de Ikkonos: https://www.ikkonos.com/ideate/que-es-el-diseno-web

Román, J. (2019). *Lenguaje CSS*. Obtenido de https://lenguajecss.com/

Rumbaugh, M., Jacobson, I., & Booch, G. (2000). *EL LENGUAJE UNIFICADO DE MODELADO.* Madrid: Pearson Educación. S.A.

Silberschatz, A., F. Korth, H., & Sudarshan, S. (2006). *Fundamentos de diseño de bases de datos Quinta Edición* . México: McGrawHill.

Sommerville, I. (2005). Ingeniería de software. En I. Sommerville, *Ingeniería de software* (pág. 712). Madrid: Pearson Addison Wesley.

Valdés, D. P. (3 de Julio de 2007). *Maestros del Web*. Obtenido de http://www.maestrosdelweb.com/que-es-javascript/

Vue.js. (2019). *Vue.js*. Obtenido de https://vuejs.org/v2/guide/

Web, D. (14 de Septiembre de 2018). *desarrolloweb.com*. Obtenido de Librería Axios: cliente HTTP para Javascript: https://desarrolloweb.com/articulos/axios-ajax-cliente-http-javascript.html

9. GLOSARIO

Software: Programas de ordenador y la documentación asociada. Los productos de software se pueden desarrollar para algún cliente en particular o para un mercado general (Sommerville, 2005, pág. 5).

Framework: Conjunto de librerías y patrones que permiten realizar proyectos con una estructura organizada y dividida, que permiten una mayor agilidad a la hora de programar.

Boostrap: Conjunto de herramientas de diseño, que permiten agilizar el maquetado y desarrollo de interfaces.

Front-End: Parte gráfica y visual del sistema, donde se manipulan los diferentes módulos por parte del usuario o cliente final.

Back-End: Lógica y estructura base de la aplicación, codificación orientada al lado del cliente, donde se crean las instrucciones necesarias para realizar una funcionalidad

Open Source: Herramientas de código abierto, con licencias libres de cambios y acceso a modificar el código fuente de cualquier herramienta.

PHP: Lenguaje de programación del lado del servidor y usado para aplicaciones web dinámicas.

VUEJS: Framework de programación de la del usuario donde se realizan las interfaces y funcionalidades previas al consumo del backend.

Casos de uso: Descripción técnica y detallada de como realizara las funciones o acciones un módulo o actividades específicas.

Iteración: Proceso por el cual se repiten varias veces dentro de un ciclo o desarrollo de un proceso de software.

RUP: Es un proceso iterativo para la construcción de software, dado a su flexibilidad de adaptación a proyectos de software y su fácil implementación con avances y pruebas de cada módulo.