



Presentado ante la ilustre Universidad de Los Andes como requisito final para
obtener el Título de Ingeniero de Sistemas

Desarrollo de aplicación web para registro académico y administrativo de los estudiantes de la SEPEC

Por

Br. Abraham Jesús Mousalli Zamora

Tutor: Dr. Albio Marquez

© 2024 Universidad de Los Andes, Mérida, Venezuela.

C.C. Reconocimiento

Desarrollo de aplicación web para registro académico y administrativo de los estudiantes de la SEPEC

Br. Abraham Jesús Mousalli Zamora

Proyecto de grado – Sistemas Computacionales, 146 páginas.

Resumen: En la actualidad, distintas dependencias y facultades de la Universidad de los Andes (ULA) en su sede en Mérida, carecen de recursos tecnológicos actualizados para realizar sus actividades administrativas y estudiantiles. El propósito de esta propuesta es abordar el problema del registro y del control académico y administrativo de cada estudiante de la Sección de Estudios de Posgrado en Economía (SEPEC), unidad académica adscrita al Instituto de Investigaciones Económicas y Sociales de la Universidad de Los Andes (ULA). Se utilizarán técnicas de recolección de datos como la entrevista y la observación directa; a partir de esta recolección de información, se diseñarán los distintos módulos y los datos que manipulará el sistema. Se empleará una metodología de desarrollo de *software* SCRUM para la planeación y para el desarrollo de una solución tecnológica que satisfaga el propósito de este proyecto. En este sentido, se busca obtener una aplicación web desarrollada en el lenguaje Python y con una base de datos propia que permita registrar estudiantes, asociar pagos y cuentas por pagar, registrar notas y ejecutar otras funcionalidades con la finalidad de centralizar la información de los estudiantes adscritos a algún programa de formación de la SEPEC y de mejorar los procedimientos internos relacionados con la secretaria de dicha unidad académica. En esta propuesta de proyecto de grado, se presenta el problema que se va a desarrollar, los objetivos que se quieren alcanzar, la metodología a seguir, los antecedentes del problema y los cronogramas de evaluación y de actividades.

Palabras clave: Aplicación web, Sistema de información, Sistema de registro y control estudiantil, Python, Base de datos.

Índice General

Índice de Figuras	XI
Índice de Tablas	XIII
Agradecimientos	XIV
Capítulo I Introducción	1
1.1 Planteamiento del problema.....	2
1.2 Delimitación del problema.....	4
1.3 Objetivos.....	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos.....	4
1.4 Metodología.....	5
1.5 Ámbito de aplicación.....	6
1.6 Justificación.....	6
Capítulo II Marco Teórico	9
2.1 Antecedentes De La Investigación.....	9
2.2 Sistema.....	12
2.2.1 Definición.....	12
2.2.2 Sistema De Información.....	13
2.2.3 Sistema de Inscripción.....	13
2.2.3.1 Gestión Académica.....	13
2.2.4 Soluciones tecnológicas.....	14
2.3 Aplicación Web.....	14
2.3.1 Definición.....	14
2.3.2 Página Web vs. Aplicación Web.....	15
2.3.3 Características de las Aplicaciones Web.....	15

2.4 Lenguaje De Programación.....	16
2.4.1 Definición.....	16
2.4.2 Python.....	16
2.4.3 JavaScript.....	16
2.5 Framework.....	17
2.5.1 Definición.....	17
2.5.2 Django.....	17
2.5.3 HTML.....	17
2.5.4 CSS.....	18
2.5.5 Bootstrap.....	18
2.5.6 AJAX.....	18
2.6 Base De Datos.....	19
2.6.1 Definición.....	19
2.6.2 MySQL.....	19
2.6.3 Características De MySQL.....	20
2.6.4 Modelo Entidad-Relación.....	20
2.6.5 Modelo Relacional.....	20
2.6.6 CRUD.....	21
2.7 Metodologías Ágiles.....	22
2.7.1 Definición.....	22
2.7.2 Metodologías Tradicionales vs. Metodologías Ágiles.....	22
2.7.3 Metodología SCRUM.....	23
2.7.4 Fases Y Elementos De La Metodología SCRUM.....	23
Capítulo III Marco Metodológico.....	26
3.1 Tipo De Investigación.....	26

3.2 Enfoque De La Investigación.....	27
3.3 Población Y Muestra.....	27
3.3.1 Población.....	27
3.3.2 Muestra.....	28
3.3.4 Tipo De Muestra.....	28
3.4 Instrumento Para La Recolección De Datos.....	28
3.4.1 Observación Directa.....	29
3.4.2 Entrevista.....	29
3.5 Procesamiento Y Codificación De Datos.....	29
3.6 Cronograma De Actividades.....	30
3.7 Cronograma De Evaluaciones.....	31
3.8 Aplicación De La Metodología SCRUM.....	31
3.9 Desarrollo De La Metodología SCRUM.....	32
3.9.1 Alcance.....	32
3.9.2 Roles.....	33
Capítulo IV Extracción De Requerimientos.....	34
4.1 Formato De Los Requerimientos.....	34
4.1.1 Historias De Usuario.....	34
4.1.2 Usuario.....	34
4.1.3 Priorización.....	34
4.1.4 Criterios De Aceptación.....	35
4.1.5 Requisitos Funcionales.....	35
4.1.6 Limitaciones.....	35
4.1.7 Dependencia.....	35
4.1.8 Comentario.....	35

4.2 Lista De Requerimientos.....	36
4.2.1 Configuración Del Proyecto En Django.....	36
4.2.2 Crear Aplicaciones Modulares En El Proyecto Django.....	37
4.2.3 Establecer La Arquitectura De Ficheros Del Proyecto Django.....	38
4.2.4 Configurar El Archivo settings.py Del Proyecto Django.....	39
4.2.5 Configuración Inicial De La Base De Datos.....	40
4.2.6 Sistema De Login.....	41
4.2.7 Crear Un Nuevo Estudiante.....	42
4.2.8 Actualizar Información Del Estudiante.....	43
4.2.9 Eliminar Estudiante.....	44
4.2.10 Visualizar Estudiante.....	45
4.2.11 Crear Una Carrera Nueva.....	46
4.2.12 Actualizar Una Carrera.....	47
4.2.13 Crear Una Cohorte.....	47
4.2.14 Calcular Fecha Posible De Culminación.....	48
4.2.15 Definir Un Semestre Actual.....	49
4.2.16 Actualizar Fechas Del Semestre.....	50
4.2.17 Registrar Una Asignatura.....	50
4.2.18 Asociar Una Asignatura A Una Carrera.....	51
4.2.19 Crear Un Plan De Estudio.....	52
4.2.20 Actualizar Un Plan De Estudio.....	53
4.2.21 Asignar Una Asignatura A Un Semestre Y A Un Plan De Estudio.....	54
4.2.22 Actualizar Asignatura Asociada A Un Plan De Estudio.....	54
4.2.23 Matricular Un Estudiante.....	55
4.2.24 Actualizar El Estatus Del Estudiante En Su Matriculación.....	56

4.2.25 Registrar La Inscripción De Un Estudiante En Una Materia.....	57
4.2.26 Actualizar La Inscripción De Un Estudiante.....	58
4.2.27 Visualizar Todas Las Inscripciones De Un Estudiante.....	59
4.2.28 Registrar Calificación De Una Asignatura.....	59
4.2.29 Actualizar Una Calificación Existente.....	60
4.2.30 Visualizar El Historial De Calificaciones De Un Estudiante.....	61
4.2.31 Registrar Un Nuevo Arancel, Un Curso O Una Unidad Curricular.....	62
4.2.32 Actualizar Un Nuevo Arancel, Un Curso O Una Unidad Curricular.....	63
4.2.33 Registrar El Precio De Un Arancel Para Una Nacionalidad Específica....	64
4.2.34 Actualizar El Precio De Un Arancel Para Una Nacionalidad Específica...	64
4.2.35 Registrar El Precio De Un Curso Para Una Nacionalidad Específica.....	65
4.2.36 Actualizar El Precio De Un Curso Para Una Nacionalidad Específica.....	66
4.2.37 Registrar El Precio De Una Unidad Curricular Para Una Nacionalidad Específica.....	67
4.2.38 Actualizar El Precio De Una Unidad Curricular Para Una Nacionalidad Específica.....	67
4.2.39 Crear Una Nueva Factura Para Un Estudiante.....	68
4.2.40 Agregar Un Elemento A Una Factura Existente.....	69
4.2.41 Registrar Un Pago Para Una Factura.....	70
4.2.42 Generar Un Reporte De Calificaciones De Un Estudiante Por Semestre...	71
4.2.43 Generar Un Reporte De Pagos Efectuados En Un Mes Específico.....	72
4.2.44 Generar Un Reporte De Estudiantes Matriculados Por Carrera.....	73
4.2.45 Generar Un Reporte De Todas Las Facturas.....	74
4.3 Product Backlog.....	75
4.4 Sprint Backlog.....	81

Capítulo V Propuesta De Solución.....	84
5.1 Arquitectura A Utilizar.....	84
5.2 Herramientas A Utilizar.....	85
5.2.1 Django.....	86
5.2.2 Aplicación de Django.....	87
5.2.3 Visual Studio Code.....	87
5.2.4 MySQL Workbench.....	87
5.2.5 AJAX.....	88
Capítulo VI Análisis De Resultados.....	89
6.1 Sprint N.º 1.....	89
6.1.1 Historia De Usuario #3: Establecer la arquitectura de ficheros del proyecto Django.....	90
6.1.2 Historia De Usuario #5: Configuración inicial de la Base de Datos.....	91
6.2 Sprint N.º 2.....	93
6.2.1 Historia De Usuario #6: Sistema De Login.....	93
6.2.2 Historia De Usuario #7: Crear Un Nuevo Estudiante.....	94
6.2.3 Historia De Usuario #8: Actualizar Información Del Estudiante.....	95
6.2.4: Historia De Usuario #10: Visualizar Estudiante.....	96
6.3 Sprint N.º 3.....	97
6.3.1: Historia De Usuario #11: Crear Una Carrera Nueva.....	98
6.3.2: Historia De Usuario #12: Actualizar Una Carrera.....	99
6.3.3: Historia De Usuario #13: Crear Una Cohorte.....	100
6.3.4: Historia De Usuario #14: Calcular Fecha Posible De Culminación.....	100
6.3.5: Historia De Usuario #15: Definir Un Semestre Actual.....	101
6.3.6: Historia De Usuario #16: Actualizar Fechas Del Semestre.....	101
6.3.7: Historia De Usuario #17: Registrar Una Asignatura.....	102

6.3.8: Historia De Usuario #18: Asociar Una Asignatura A Una Carrera.....	103
6.3.9: Historia De Usuario #19: Crear Un Plan De Estudio.....	103
6.3.10: Historia De Usuario #20: Actualizar Un Plan De Estudio.....	104
6.3.11: Historia De Usuario #21: Asignar Una Asignatura A Un Semestre Y A Un Plan De Estudio.....	105
6.3.12: Historia De Usuario #22: Actualizar Asignatura Asociada A Un Plan De Estudio.....	106
6.3.13: Historia De Usuario #23: Matricular Un estudiante.....	106
6.3.14: Historia De Usuario #24: Actualizar El Estatus Del Estudiante En Su Matriculación.....	108
6.4 Sprint N.º 4.....	109
6.4.1: Historia de Usuario #25: Registrar La Inscripción De Un Estudiante En Una Materia.....	109
6.4.2: Historia De Usuario #26: Actualizar La Inscripción De Un Estudiante...	110
6.4.3: Historia De Usuario #27: Visualizar Todas Las Inscripciones De Un Estudiante.....	110
6.4.4: Historia De Usuario #28: Registrar Calificación De Una Asignatura.....	111
6.4.5: Historia De Usuario #29: Actualizar Una Calificación Existente.....	112
6.4.6: Historia De Usuario #30: Visualizar El Historial De Calificaciones De Un Estudiante.....	113
6.5 Sprint N.º 5.....	113
6.5.1: Historia De Usuario #31: Registrar Un Nuevo Arancel, Un Curso O Una Unidad Curricular.....	114
6.5.2: Historia De Usuario #32: Actualizar Un arancel, Un curso O Una Unidad Curricular.....	115
6.5.3: Historia De Usuario #33: Registrar El Precio De Un Arancel Para Una Nacionalidad Específica.....	115
6.5.4: Historia De Usuario #34: Actualizar El Precio De Un Arancel Para Una Nacionalidad Específica.....	116

6.6 Sprint N.º 6.....	116
6.6.1: Historia De Usuario #39: Crear Una Nueva Factura Para Un Estudiante.....	117
6.6.2: Historia De Usuario #40: Agregar Un Elemento A Una Factura Existente.....	118
6.6.3: Historia De Usuario #41: Registrar Un Pago Para Una Factura.....	120
6.7 Sprint N.º 7.....	120
6.7.1: Historia De Usuario #42: Generar Un Reporte De Calificaciones De Un Estudiante Por Semestre.....	121
6.7.2: Historia De Usuario #43: Generar Un Reporte De Pagos Efectuados En Un Mes Específico.....	122
6.7.3: Historia De Usuario #44: Generar Un Reporte De Estudiantes Matriculados Por Carrera.....	123
6.7.4: Historia De Usuario #45: Generar Un Reporte De Todas Las Facturas...	124
Capítulo VII Conclusiones Y Recomendaciones.....	125
7.1 Conclusiones.....	125
7.2 Recomendaciones.....	126
Referencias Bibliográficas.....	127

Índice De Figuras

Figura 1 Esquema básico de una aplicación web. Fuente: (Luján Mora, 2002).....	15
Figura 2 Visualización del <i>framework</i> SCRUM. Fuente: Project Management.....	25
Figura 3 Estructura de ficheros del proyecto Django. Fuente: Propia.....	90
Figura 4 Diagrama Entidad-Relación del modelo de la base de datos. Fuente: Propia.....	92
Figura 5 Inicio de sesión en la aplicación web. Fuente: Propia.....	94
Figura 6 Notificación cuando se ingresa un dato incorrecto en el inicio de sesión. Fuente: Propia.....	94
Figura 7 Formulario para crear un nuevo estudiante. Fuente: Propia.....	95
Figura 8 Notificación de error tras intentar crear un usuario con CI/Pasaporte ya existente. Fuente: Propia.....	96
Figura 9 Buscador de estudiantes por número de CI o Pasaporte. Fuente: Propia.....	96
Figura 10 Visualizar los datos almacenados del estudiante. Fuente: Propia.....	97
Figura 11 Formulario para crear un programa o una carrera. Fuente: Propia.....	99
Figura 12 Formulario para actualizar un programa o una carrera. Fuente: Propia.....	99
Figura 13 Formulario para crear una cohorte. Fuente: Propia.....	100
Figura 14 Formulario para actualizar una cohorte. Fuente: Propia.....	100
Figura 15 Método <i>save</i> sobrescrito para calcular la posible fecha de finalización. Fuente: Propia.....	101
Figura 16 Formulario para crear un semestre. Fuente: Propia.....	101
Figura 17 Formulario para actualizar un semestre. Fuente: Propia.....	102
Figura 18 Formulario para crear una asignatura. Fuente: Propia.....	102
Figura 19 Formulario para actualizar una asignatura. Fuente: Propia.....	103
Figura 20 Formulario para crear un plan de estudio. Fuente: Propia.....	104
Figura 21 Formulario para actualizar un plan de estudio. Fuente: Propia.....	104
Figura 22 Interfaz de plan de estudio y de las asignaturas que tiene asociada. Fuente: Propia...	105

Figura 23	Formulario para asociar una asignatura a un plan de estudio. Fuente: Propia.....	106
Figura 24	Edición de la etapa del semestre de una asignatura en un plan de estudio. Fuente: Propia.....	106
Figura 25	Submenú dentro del módulo de un estudiante existente.....	107
Figura 26	Interfaz para matricular un estudiante por primera vez.....	107
Figura 27	Formulario para matricular a un estudiante en un plan de estudio. Fuente: Propia...	108
Figura 28	Formulario para actualizar la matriculación de un estudiante. Fuente: Propia.....	108
Figura 29	Formulario para inscribir una asignatura en un semestre a un estudiante. Fuente: Propia.....	110
Figura 30	Formulario para actualizar la inscripción de un estudiante en un semestre activo. Fuente: Propia.....	110
Figura 31	Interfaz para visualizar las asignaturas inscritas en un semestre activo. Fuente: Propia.....	111
Figura 32	Interfaz para visualizar las asignaturas inscritas y para poder asignar una calificación. Fuente: Propia.....	111
Figura 33	Formulario para registrar una calificación a una asignatura inscrita. Fuente: Propia.....	112
Figura 34	Formulario para actualizar la calificación a una asignatura inscrita. Fuente: Propia..	112
Figura 35	Formulario para actualizar la calificación a una asignatura inscrita. Fuente: Propia..	113
Figura 36	Formulario para crear un arancel. Fuente: Propia.....	114
Figura 37	Formulario para actualizar un arancel. Fuente: Propia.....	115
Figura 38	Formulario para crear el precio de un arancel para una nacionalidad específica. Fuente: Propia.....	115
Figura 39	Formulario para actualizar el precio de un arancel para una nacionalidad específica. Fuente: Propia.....	116
Figura 40	Formulario para crear una nueva factura a un estudiante. Fuente: Propia.....	118
Figura 41	Interfaz de una factura asociada a un estudiante. Fuente: Propia.....	119
Figura 42	Formulario para facturar un producto a un estudiante. Fuente: Propia.....	119
Figura 43	Formulario para registrar un pago a la factura de un estudiante. Fuente: Propia.....	120

Figura 44 Reporte PDF de calificaciones registradas a un estudiante por semestre. Fuente: Propia.....	122
Figura 45 Reporte PDF para listar los pagos recibidos para un mes/año seleccionado. Fuente: Propia.....	123
Figura 46 Reporte PDF de la lista de estudiantes matriculados a un plan de estudio. Fuente: Propia.....	123
Figura 47 Reporte Excel de las facturas creadas. Fuente: Propia.....	124

www.bdigital.ula.ve

Índice De Tablas

Tabla 1 Cuadro comparativo en donde se relacionan los procesos tradicionales vs. SIISBC. Fuente: Lijeron (2009).....	10
Tabla 2 Descripción de productos y formalismos según la metodología OOHDM. Fuente: Urbano y Gómez (2022).....	12
Tabla 3 Metodologías tradiciones vs. metodologías Ágiles. Fuente: Cadavid, Martínez y Vélez (2013).....	22
Tabla 4 Cronograma de Actividades. Fuente: Propia.....	31
Tabla 5 Cronograma de evaluaciones. Fuente: Propia.....	31
Tabla 6 Product Backlog Fuente: Propia.....	76
Tabla 7 Product Backlog Fuente: Propia.....	77
Tabla 8 Product Backlog Fuente: Propia.....	78
Tabla 9 Product Backlog Fuente: Propia.....	79
Tabla 10 Product Backlog Fuente: Propia.....	80
Tabla 11 Product Backlog Fuente: Propia.....	81
Tabla 12 Sprint Backlog Fuente: Propia.....	82
Tabla 13 Sprint Backlog Fuente: Propia.....	83
Tabla 14 Historias de usuario incorporadas en el Sprint N.º 1. Fuente: Propia.....	90
Tabla 15 Historias de usuario incorporadas en el Sprint N.º 2. Fuente: Propia.....	93
Tabla 16 Historias de usuario incorporadas en el Sprint N.º 3. Fuente: Propia.....	98
Tabla 17 Historias de usuario incorporadas en el Sprint N.º 4. Fuente: Propia.....	109
Tabla 18 Historias de usuario incorporadas en el Sprint N.º 5. Fuente: Propia.....	114
Tabla 19 Historias de usuario incorporadas en el Sprint N.º 6. Fuente: Propia.....	117
Tabla 20 Historias de usuario incorporadas en el Sprint N.º 7. Fuente: Propia.....	121

Capítulo I

Introducción

Los sistemas de información administrativa permiten automatizar y aumentar la eficiencia a la hora de realizar trámites y procesos administrativos. Lo mismo sucede al llevar el seguimiento y el control estudiantil en instituciones dedicadas a prestar servicios de educación. De esta necesidad, no escapan las universidades, quienes, hoy en día, requieren de soluciones tecnológicas eficientes y personalizadas para trabajar organizadamente; en especial, en las áreas relacionadas con los estudiantes inscritos en algún programa de formación. Desde el momento de inscripción hasta su egreso, la universidad necesita administrar la información académica y administrativa de cada estudiante.

www.bdigital.ula.ve

La pregunta principal en esta investigación es como mejorar el registro académico y administrativo de los estudiantes inscritos en los programas de formación de la SEPEC. La respuesta a esta interrogante se asocia con llevar adelante un proyecto cuyo objetivo central es desarrollar una aplicación web con el fin de mejorar dicho registro académico y administrativo.

A fines profesionales, el desarrollo de este proyecto permitió tener un acercamiento con el desarrollo de software, entendiendo las situaciones reales y buscando la mejor solución para poder abordar el problema. Al mismo tiempo, contribuyó a seguir adquiriendo nuevos conocimientos que, en el ámbito laboral, son necesarios dominar.

1.1. Planteamiento Del Problema

La Sección de Estudios de Posgrado en Economía (SEPEC)¹ está adscrita al Instituto de Investigaciones Económicas y Sociales de la Universidad de Los Andes (ULA). Inició en el primer semestre del año 1995 con el Programa de Maestría en Economía. Desde entonces, ha logrado incluir nuevos programas de formación en Maestría, en Doctorado y, desde hace poco, en cursos acreditables cortos. Se encuentra ubicada en la ciudad de Mérida, Facultad de Ciencias Económicas y Sociales (FACES-ULA), Núcleo La Liria, edificio G. “Leocadio Hontoria”, 3.^{er} piso.

Desde su comienzo, el registro académico y administrativo ha sido realizado de una manera clásica y rudimentaria. Esto se debe a la falta de interés por aplicar y sacar provecho de soluciones tecnológicas más eficientes. En la actualidad, acceder a la información es vital para poder tomar decisiones estratégicas, para brindar una atención personalizada, para listar cualquier duda proveniente de un usuario, entre otros. Esto permite un mejor control de los procedimientos y de los procesos ante la gestión académica y administrativa en cualquier institución.

En la SEPEC, para tener información inmediata en cuanto a los recursos financieros, al registro y al control de la información de un estudiante, a la gestión de los pagos de aranceles y otro sin fin de procesos, continúa realizándose un proceso de manera empírica, manual y sin algún programa o software.

¹ Instituto de Investigaciones Económicas y Sociales <http://iies.faces.ula.ve/sepec/maestria.html>.

La falta de soluciones tecnológicas incorporadas en sus procedimientos académicos y administrativos hace que no exista una más eficiente gestión de la información. No se tiene especificado ningún tipo de manejo eficiente y control de la administración, salvo utilizar hojas en el programa Excel de manera local. Dado este panorama, es de esperarse que los procesos de registro académico y administrativos de esta unidad den como resultado retrasos, inconvenientes y un esfuerzo enorme al momento de procesar los balances del saldo actual y cuentas por cobrar o por pagar. Para algunos cálculos financieros y administrativos, se obtiene información de una manera manual y esto incrementa las probabilidades de errores humanos; encima, no hay recursos financieros para contratar un proveedor o una solución tecnológica externa.

El registro estudiantil es otro factor que no se lleva con un control eficiente, ya que no se apunta el tiempo determinado por la institución para que un estudiante complete su programa de formación. Se espera que, en un futuro cercano, la plantilla de estudiantes inscritos en algún programa de formación se incremente y el no tener una solución tecnológica podría perjudicar el control y el avance de los estudiantes en dichos programas.

Por tanto, en esta propuesta, se buscó dar una solución al problema actual de la SEPEC, es decir, la digitalización de sus procesos académicos y administrativos. Para hacer esto posible, se desarrolló una solución tecnológica que tenga una interfaz web con un almacenamiento local. Se planteó que, a través de los siguientes módulos, se puedan dividir sus procedimientos: módulo de estudiante, módulo de registro académico, módulo de plan de formación, módulo de facturas, módulo de productos y módulo de reportes en formato .pdf y .xlsx. Esto permitió a la SEPEC digitalizar la información y obtener, en tiempo real, detalles de los estudiantes como estatus de pagos pendiente, materias cursadas, entre otros.

1.2. Delimitación Del Problema

En este proyecto, se planteó:

- ¿Cómo agilizar el registro académico y administrativo de los estudiantes inscritos en los programas de formación de la SEPEC?
- ¿Existe una manera de obtener información precisa e inmediata sobre el estatus académico de un estudiante inscrito a un programa de formación de la SEPEC?

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una aplicación web con el fin de mejorar el registro académico y administrativo de los estudiantes inscritos en los programas de formación de la SEPEC.

1.3.2. Objetivos Específicos

1. Diagnosticar los procesos académicos y administrativos de la SEPEC mediante una serie de entrevistas con el personal administrativo de la unidad académica.
2. Determinar los requerimientos y las necesidades a satisfacer agrupándolos en un listado.
3. Identificar los recursos tecnológicos disponibles para el diseño y para el desarrollo de una aplicación web que afronte o resuelva los requerimientos y las necesidades actuales.
4. Diseñar una aplicación web interactiva, con buena experiencia de usuario y de fácil uso.
5. Desarrollar una aplicación web que permita el registro académico y administrativo de los estudiantes inscritos en los programas de formación de la SEPEC.

1.4. Metodología

Para esta investigación, se aplicaron técnicas, principios y fundamentos del proyecto factible. La UPEL (2003) define el proyecto factible como “un estudio que consiste en la investigación, en la elaboración y en el desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales” (p.16).

La técnica de entrevista fue de gran valor para poder entender las necesidades y para diagnosticar el problema. Según Dubs (2004), “se hace referencia a la entrevista de investigación, en la cual el sujeto constituye la fuente de la información.” (p. 94). Además, se emplearon los principios y los fundamentos de una metodología de desarrollo de *software*, la cual permitió la elaboración del producto utilizando la metodología o *framework* SCRUM, que es una de las metodologías ágiles. Rad y Turley (2019) definen el término *Agile* como “el nombre más común para los sistemas que utilizan los ciclos de vidas adaptativos” (p. 5). Existen varios tipos de metodologías ágiles y, para Rodríguez y Dorado (2015), “Scrum es una de las metodologías de desarrollo ágil de Software más reconocidas a nivel mundial” (p. 133). Rodríguez y Dorado (2015) mencionan también que SCRUM “permite que exista una relación presente e interactiva entre el cliente en conjunto con un equipo de desarrollo, definiendo entre ambos qué se hace y cómo se hace con base en una duración del *sprint* (ciclo de desarrollo) e involucrando al cliente en la inspección” (p. 135).

Al aplicar esta metodología de *software*, junto con el cronograma de actividades, se desarrolló esta aplicación web cumpliendo los objetivos específicos antes definidos y las necesidades actuales.

1.5. Ámbito De Aplicación

La Sección de Estudios de Posgrado en Economía, en sus siglas SEPEC, está adscrita al Instituto de Investigaciones Económicas y Sociales de la Universidad de Los Andes desde el primer semestre del año 1995 con el Programa de Maestría en Economía. Actualmente, dentro de sus ofertas académicas, se incluye un Doctorado en Economía Aplicada, cursos cortos y 5 menciones de la Maestría en Economía (Economía Cuantitativa, Políticas Económicas, Política Fiscal y Tributaria, Políticas Públicas y Territoriales, y Economía y Políticas Agroalimentarias).

Este proyecto recopila información de datos académicos, administrativos y personales de los estudiantes, el nombre y el código de cada asignatura por la mención de maestría o del doctorado, la información de la relación de pagos de los estudiantes, el estatus de procesos de inscripción de cada estudiante y las notas de las asignaturas inscritas por estudiante del programa de formación inscrito en la SEPEC.

1.6. Justificación

López (2021) menciona un aspecto clave sobre la tecnología, más específicamente, sobre la digitalización, definiéndola como “un proceso que ha surgido en el mundo desde décadas atrás, desde la invención de la primera computadora electrónica en 1939 por el estadounidense John Atanasoff” (p. 2). En las instituciones educativas venezolanas, la inclusión de la tecnología y la digitalización de sus procesos ha sido implementada de diferentes maneras y, en algunos casos, ciertas dependencias, áreas o procedimientos siguen siendo atendidas de manera física y manual aún.

Respecto a la gestión académica, Inciarte, Marcano y Reyes (2006) comentan que es “un recorrido que involucra las acciones de atención al estudiante desde el ingreso hasta su desarrollo y egreso del sistema”. Digitalizar e implementar tecnología permite un fácil acceso de la información y, en tiempo real, obtener estatus, tiempos, solicitudes, entre otros. Es por eso que el propósito de la siguiente investigación fue diseñar y desarrollar una solución tecnológica que sea de fácil uso, en un entorno web y con un almacenamiento local de la SEPEC.

Tener acceso al expediente y obtener información precisa permite facilitar el trabajo diario del personal administrativo y directivo de la SEPEC; especialmente, en un proceso de inscripción. Huete y Carrillo (2017) comentan que, si existe un proceso de inscripción electrónico en una institución, este “proporciona información clara y precisa, agiliza el proceso de inscripción, procesa la información en período de tiempo corto, permite obtener información completa sobre trabajos terminados” (p. 3). Es por eso que la presente investigación se realizó con el fin de mejorar el registro académico y administrativo de los estudiantes inscritos a algún programa de formación de la SEPEC, ayudando al personal directivo y administrativo a digitalizar sus procesos, a incluir tecnología, a evitar la pérdida de información académica de algún estudiante y a acceder a información financiera de una manera más simple.

De hecho, para que un estudiante complete un estudio de formación en Maestría o Doctorado en la SEPEC, se requiere cumplir una serie de requisitos, desde la mención a la que pertenece, tiempo en el programa, asignaturas aprobadas, calificaciones, entre otros. Para Olivares, Meza y Garrido (2018), existe una necesidad respecto a los procesos de inscripción y el control sobre las asignaturas a cursar en un periodo; en sus conclusiones, mencionan que “los estudiantes no poseen herramientas en las que apoyarse para tomar buenas decisiones en la

inscripción de módulos”. En este caso, la SEPEC requiere tener un control sobre las asignaturas cursadas por un estudiante inscrito en algún programa de formación.

Con ayuda de 2 tipos de metodología, una de investigación, utilizando principalmente los pilares o los fundamentos del proyecto factible, y otra de desarrollo de *software*, implementando el tipo de metodología ágil llamada Scrum, se desarrolló este proyecto. Ambas permitieron el diseño y el desarrollo de una solución tecnológica que pueda satisfacer las necesidades planteadas en dicha propuesta.

En el ámbito organizacional, este proyecto justificó su causa y desarrollo porque la SEPEC está en búsqueda de un procedimiento que eleve su productividad y permita digitalizar la información para, con ello, ahorrar el tiempo de consultas y administrar mejor sus recursos. Se estaría aportando una alternativa de solución para problemas con una misma situación o realidad. Además, este proyecto servirá de antecedentes a otros investigadores que se enfrenten a un conjunto de problemáticas y requerimientos similares a los aquí propuestos.

Capítulo II

Marco Teórico

En este capítulo, se desarrollan los antecedentes de investigación, los cuales tienen la funcionalidad de aportar bases y conocimientos respecto a soluciones tecnológicas, aplicaciones web, sistemas de inscripción y registro de pagos académicos. Se presentan las definiciones de sistema de gestión académica, de sistema de inscripción, de sistema de registro de pagos, de soluciones tecnológicas, y de aplicación web. A su vez, se especifican las funcionalidades de los entornos de trabajo, de lenguajes y de *frameworks* como Python, Django, HTML, CSS, Bootstrap, MySQL y otros, así como una descripción de la metodología ágil SCRUM. De esta manera, se brindan definiciones y conceptos que permiten al lector tener una noción de lo expuesto para facilitar su comprensión del contenido en lo que resta de este proyecto.

2.1 Antecedentes De La Investigación

Desde la inclusión de la tecnología en el ámbito universitario, los procedimientos y todo lo referente a procesos académicos y administrativos han evolucionado y se han adaptado a las nuevas realidades que se viven en esta era digital. Existe una gran variedad de soluciones tecnológicas implementadas y desarrolladas para abordar procesos de inscripción o de matriculación, pagos, aranceles y otras solicitudes.

Como señalan García, Argueta y Hernández (2006), para finales de los años 90, ya existe la necesidad de implementar tecnología a los procesos de inscripción y registro académico en etapa universitaria. Al cambiar procesos tradicionales, es necesario ofrecer material de ayuda y de procedimientos para facilitar el uso correcto de cualquier solución tecnológica implementada.

Tener implementada alguna solución tecnológica personalizada para el registro académico y administrativo da múltiples beneficios: permite agilizar tareas, estandarizar procedimientos, ahorrar costos de material de oficina, mejorar la efectividad de los empleados y del departamento, entre otros aspectos positivos.

Para los antecedentes de la investigación, que “reflejan los avances y el estado actual del conocimiento en un área determinada, y sirven de modelo o de ejemplo para futuras investigaciones.” (Arias, 2012), se presentan algunos proyectos relacionados con el tema de investigación.

Lijeron (2009) menciona que, con ayuda de una metodología de *software* del modelo clásico, se logró resolver una necesidad real, teniendo diferentes módulos para gestionar solicitudes distintas, pero, en algunos casos, asociadas o relacionadas entre sí. En la Tabla #1, se observa un cuadro comparativo en donde se relacionan los procesos tradicionales realizados hasta antes del desarrollo de la solución en relación con la implementación de la nueva solución tecnológica personalizada con base en los requerimientos y en las necesidades detectadas.

Sistema Tradicional	SIISBC
Información Incompleta	Posee la información necesaria
Manipulación de datos lenta	Manipulación de datos rápida
Sin registro de los miembros	Registro de los miembros
Sin el control de comunidades	Control de comunidades existentes
Perdida en el registro de préstamo de material de formación	Ágil registro de préstamo de material de formación
Perdida en el registro de devolución de material de formación	Ágil registro de devolución de material de formación
Se desconoce el material existente	Registro y consulta del material existente en biblioteca
Pérdida de tiempo en la realización del informe económico en forma manual	Se generan los informes de forma automática y es al instante
Errores en los informes económicos	Precisión en los informes económicos

Tabla #1: Cuadro comparativo en donde se relacionan los procesos tradicionales vs. SIISBC. Fuente: Lijeron (2009)

La digitalización de la información permite un mejor control de los datos del estudiante, el acceso inmediato y la disponibilidad de la información. Ulloa (2014) utilizó diferentes tipos de instrumentos de investigación mediante encuestas a los estudiantes para mejorar el procedimiento de matriculación, empleando los principios del proyecto factible. Dentro de sus conclusiones, destaca que la creación de una base de datos eficiente permite tener una mejor conexión con la información, señalando que, “a través del esquema de base de datos establecido, se logró validar la información de los estudiantes inscritos.” La arquitectura empleada, MVC, permitió modularizar la solución tecnológica desarrollada.

En Nicaragua, Averruz, Sevilla y Martínez (2019) emplearon las entrevistas al personal directivo de la institución como técnica de recolección de datos, con el fin de reunir información relacionada con los procesos y con los comportamientos referentes al proceso de inscripción y de control administrativo para realizar un análisis de requerimientos más consistente. Gracias al uso de esta técnica, se logró desarrollar una solución tecnológica para llevar un registro académico y administrativo, y modularizar el sistema en distintas secciones.

Urbano y Gómez (2022) comentan que “para la conclusión del sistema web en tiempo y en forma, se optó por utilizar el marco de trabajo Scrum junto con la metodología de software web OOHDM”, lo que permite concluir que es posible incluir más de una metodología de *software*. En la Tabla #2, se observa una descripción de los productos y de los formalismos cuando se utiliza la metodología OOHDM.

Etapas	Descripción	Productos	Formalismos
Obtención de requerimientos	Identificar roles y tareas, especificación de escenarios, casos de uso, especificación de UID y validación de casos de uso y UIDs.	Casos de Uso (actores, escenarios).	Plantillas de formato del documento, Diagramas de Interacción de Usuario (UIDs).
Diseño Conceptual	Genera un modelo conceptual basado en los diagramas de clases y secuenciales.	Clases, subsistemas, relaciones, atributos	Modelo Orientados a Objetos
Diseño Navegacional	Desarrolla una topología navegacional que permite a la aplicación ejecutar todas las tareas requeridas por el usuario.	Nodos, enlaces, estructuras de acceso, contextos, navegacionales.	Vistas Orientadas a Objetos, Cartas de navegación orientadas a objetos, Clases de contexto.
Diseño de interfaz Abstracta	Especifica las diferentes interfaces de la aplicación; para definir de qué manera aparecerán los objetos navegacionales en la interfaz y cuales objetos activarán la navegación.	Objetos de la interfaz abstracta respuestas a eventos externos, transformaciones de la interfaz.	Vistas Abstractas de Datos (ADV), Diagramas de Configuración, Cartas de navegación de los ADV's.
Implementación	Una vez realizadas las etapas anteriores, se tiene un completo conocimiento del dominio del problema.	Aplicación en funcionamiento.	Los soportados por el entorno.

Tabla#2: Descripción de productos y formalismos según la metodología OOHDM. Fuente: Urbano y Gómez (2022).

Asimismo, Macías y Mindiola (2018) mencionan que, aplicando la metodología Scrum, “se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto” (p. 4). También, es posible utilizar listas priorizadas e historias de usuario con los objetivos y con los requisitos que se deben proceder para tener un mejor control del desarrollo de una solución tecnológica. Dentro de sus conclusiones, lograron mejorar y optimizar los tiempos en solicitudes, dejando atrás el uso de aplicativos como Excel y Word.

2.2 Sistema

2.2.1 Definición

El concepto de sistema es fundamental en una variedad de disciplinas científicas. La comprensión de lo que constituye un sistema y cómo funciona es esencial para abordar problemas complejos y para diseñar soluciones efectivas en campos que abarcan desde la ingeniería hasta la medicina. En este contexto, es relevante definir con precisión qué entendemos por “sistema” antes de sumergirnos en el análisis de una solución tecnológica.

Arnold y Osorio (1998) definen los sistemas como “conjuntos de elementos que guardan estrechas relaciones entre sí, que mantienen al sistema directo o indirectamente unido de modo más o menos estable, y cuyo comportamiento global persigue, normalmente, algún tipo de objetivo.”

2.2.2 Sistema De Información

Sánchez (2014) señala que la definición hecha por Andreu, Ricart y Valor (1991) acerca de un sistema de información se refiere a un “conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuye, selectivamente, la información necesaria para la operación de dicha empresa y para las actividades de dirección y de control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia.”

2.2.3 Sistema De Inscripción

Álvarez (2007) explica que “los sistemas de inscripción son aplicaciones que permiten a la comunidad estudiantil de una institución de educación, generalmente de nivel superior, realizar la suscripción de las materias que serán cursadas por cada uno de sus miembros durante un periodo escolar.” Existen instituciones que cuentan con sistemas de información que hace posible que sus estudiantes se matriculen a través de internet, sin necesidad de tener que acudir físicamente a la institución a formalizar su inscripción.

2.2.3.1 Gestión Académica

Como explican Reyes y Marín (2021), este tipo de gestión:

Relaciona un conjunto de actividades de aspecto netamente académico dentro de una institución educativa en donde se realizan procesos que requieren utilizar o manipular información. Si la institución emplea o usa algún sistema de información, es posible centralizar la información y hacer procesos académicos mucho más efectivos y automatizados.

2.2.4 Soluciones Tecnológicas

Henao y Arango (2006) resaltan que las soluciones tecnológicas, o “estrategias tecnológicas”, facilitan la gestión de los activos de conocimiento al estar orientadas a la identificación de conocimientos explícitos y tácitos. Estas soluciones representan una respuesta que involucra la utilización de todos los recursos disponibles respaldada por un diseño y por un método que permiten abordar el problema de manera efectiva. Durante este proceso, se identifican los inconvenientes y se crea una solución después de analizar el objeto de estudio.

2.3 Aplicación Web

2.3.1 Definición

Según Luján (2002), una aplicación web se define como “un tipo especial de aplicación cliente/servidor en la que tanto el cliente (el navegador, el explorador o el visualizador) como el servidor (el servidor web) y el protocolo mediante el cual se comunican (HTTP) están estandarizados y no requieren ser creados por el programador de aplicaciones.” En la Figura #1, se ilustra la relación cliente-servidor a través del protocolo HTTP.

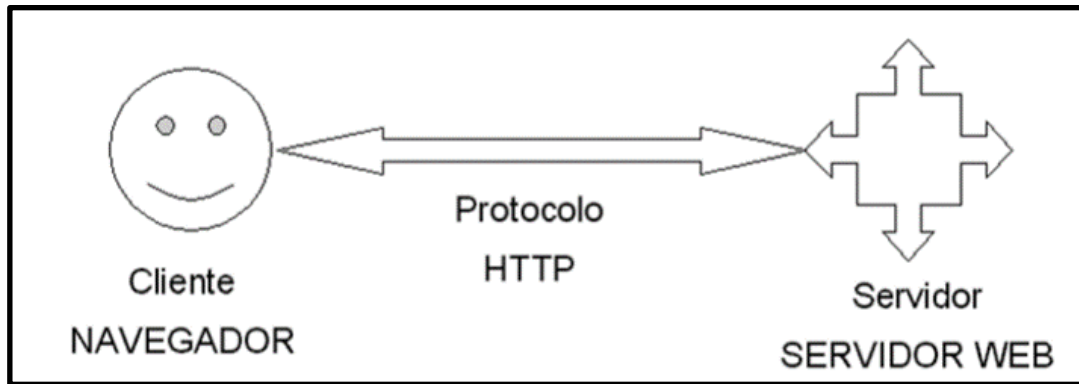


Figura #1: Esquema básico de una aplicación web. Fuente: (Luján Mora, 2002)

2.3.2 Página Web vs. Aplicación Web

De acuerdo a González (2010), se distingue una aplicación web de una página web por su capacidad de “permitir al usuario interactuar con la lógica de negocio en el servidor”, siendo la lógica de negocio la que comprende un conjunto de procesos que implementan las reglas de funcionamiento de la aplicación web. En cambio, las páginas web se centran en proporcionar información al usuario.

2.3.3 Características De Las Aplicaciones Web

Estas son algunas de las características de las aplicaciones web ² :

- Usan un único diseño, normalmente en HTML5, para todos los dispositivos.
- El usuario no tiene que descargarlas. Se alojan en un servidor y se accede a ellas desde el navegador.
- Son compatibles con cualquier navegador.

² Aplicaciones web: en qué consisten y cuáles son sus ventajas. (2020, noviembre 20). Recuperado de <https://postgradoingenieria.com/que-son-aplicaciones-web/>.

- No aparecen en los buscadores de aplicaciones como Google Play Store, pero sí en los buscadores de Internet.

2.4 Lenguaje De Programación

2.4.1 Definición

De acuerdo a lo descrito por Gómez (2012), un lenguaje de programación es “un lenguaje formal definido artificialmente para expresar algoritmos que una computadora ejecutará instrucción por instrucción.” Cada lenguaje de programación está sujeto a un conjunto de reglas sintácticas (que rigen la escritura del lenguaje) y reglas semánticas (que se aplican a las expresiones) que se utilizan para las instrucciones y para las operaciones enviadas a la computadora.

2.4.2 Python

Python ³, según la definición técnica de su página oficial, es “un lenguaje interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipado dinámico, tipos de datos de muy alto nivel y clases. Python combina un poder destacado con una sintaxis muy clara”.

2.4.3 JavaScript

JavaScript ⁴, según su página oficial en Mozilla Developer Network (MDN), se define como un lenguaje de programación de “alto nivel que se utiliza para crear contenido interactivo

³ Python Software Foundation. (s.f.). ¿Qué es Python? Recuperado de:
<https://docs.python.org/es/3/faq/general.html#what-is-python>.

⁴ Mozilla Developer Network. (s.f.). JavaScript. Mozilla Developer Network. Recuperado de:
<https://developer.mozilla.org/es/docs/Web/JavaScript>.

en sitios web. Puede funcionar en el lado del cliente (navegador del usuario) y en el lado del servidor (a través de Node.js). Es un lenguaje de *scripting* versátil y ampliamente utilizado”.

2.5 Framework

2.5.1 Definición

UNIR Revista (2022) define un *framework* como “un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y para el desarrollo de software”.

2.5.2 Django

Django ⁵, según su sitio web oficial, se define como la posibilidad de “llevar aplicaciones web desde el concepto hasta el lanzamiento en cuestión de horas. Django se encarga de gran parte de las molestias del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda”. Una de las razones destacadas de utilizar este framework es por ser gratuito y de código abierto.

2.5.3 HTML

HTML ⁶, es el lenguaje estándar para la creación de páginas web y, en su sitio web oficial, se define como “lenguaje de marcado de hipertexto. Se utiliza para crear páginas web. Es decir, las páginas web de todo el mundo están compuestas de HTML. Podemos observar cómo otras personas han codificado sus páginas web”.

⁵ The Django Project. (s.f.). What is Django? Recuperado de: <https://www.djangoproject.com/start/overview/>.

⁶ World Wide Web Consortium (W3C). (s.f.). HTML: Hypertext Markup Language. World Wide Web Consortium. Recuperado en 27 de octubre de 2023, de: https://www.w3.org/wiki/HTML/Training/What_is_HTML.

2.5.4 CSS

CSS ⁷ es un lenguaje de estilo que define la presentación de los documentos HTML. Su página de tutorial oficial en español señala que “CSS es el acrónimo de Cascading Style Sheets (es decir, hojas de estilo en cascada). Abarca cuestiones relativas a fuentes, a colores, a márgenes, a líneas, a altura, a anchura, a imágenes de fondo, a posicionamiento avanzado y a muchos otros temas”.

2.5.5 Bootstrap

Según su sitio web oficial, Bootstrap ⁸ es “un marco de diseño de código abierto que se utiliza para desarrollar sitios web y aplicaciones web. Está diseñado para ayudar a los desarrolladores a crear interfaces de usuario atractivas y responsivas utilizando HTML, CSS y JavaScript.” Es usado para el desarrollo web y se conoce por su enfoque en la creación de sitios web receptivos y para móviles también.

2.5.6 AJAX

AJAX, en sus siglas en inglés Asynchronous JavaScript and XML, es una técnica de desarrollo web que permite la comunicación asíncrona entre el navegador del usuario y el servidor web. Según IBM ⁹, AJAX es:

Un grupo de tecnologías que se utilizan para desarrollar aplicaciones web. Al combinar estas tecnologías, las páginas web parece que son más receptivas, puesto que los paquetes

⁷ HTML.net. (s.f.). Tutorial de CSS - Lección 1. HTML.net. Recuperado en 20 de enero de 2024, de: <http://es.html.net/tutorials/css/lesson1.php>.

⁸ Bootstrap. (s.f.). Recuperado en 20 de enero de 2024, de: <https://getbootstrap.com/>.

⁹ IBM. (s.f.). Asynchronous JavaScript and XML (AJAX) Overview. IBM Documentation. Recuperado en 27 de octubre de 2023, de: <https://www.ibm.com/docs/es/rational-soft-arch/9.6.1>.

pequeños de datos se intercambian con el servidor y las páginas web no se vuelven a cargar cada vez que un usuario realiza un cambio de entrada”.

2.6 Base De Datos

2.6.1 Definición

En la actualidad, todo sistema o aplicación requiere almacenar información y, de acuerdo a su enfoque, puede almacenar los datos para utilizarlos de acuerdo a las reglas de negocios. Oracle ¹⁰ define una base de datos como “una recopilación organizada de información o de datos estructurados que, normalmente, se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS)”.

2.6.2 MySQL

MySQL ¹¹ se define, según su página de documentación técnica, como “el sistema de gestión de bases de datos SQL de código abierto más popular, es desarrollado, distribuido y respaldado por Oracle Corporation”. Es un sistema de gestión de bases de datos que cuenta con múltiples características.

¹⁰ Oracle. (s.f.). ¿Qué es una base de datos? Oracle. Recuperado en 27 de octubre de 2023, de: <https://www.oracle.com/mx/database/what-is-database/>.

¹¹ MySQL. (s.f.). MySQL: The World's Most Popular Open Source Database. Recuperado en 27 de octubre de 2023, de: https://docs.oracle.com/cd/E17952_01/mysql-8.0-en/what-is-mysql.html.

2.6.3 Características De MySQL

En la documentación técnica de MySQL ¹², se mencionan algunas de las características principales de MySQL:

- MySQL es un sistema de gestión de bases de datos.
- Las bases de datos MySQL son relacionales.
- El *software* MySQL es de código abierto.
- El servidor de base de datos MySQL es muy rápido, confiable, escalable y fácil de usar.
- MySQL Server funciona en cliente/servidor o en sistemas integrados.
- Se encuentra disponible una gran cantidad de *software* MySQL contribuido.

2.6.4 Modelo Entidad - Relación

La plataforma Lucidchart ¹³ define un modelo de entidad-relación, también denominado diagrama entidad-relación o ERD, como “un tipo de diagrama de flujo que ilustra cómo las «entidades» como personas, objetos o conceptos se relacionan entre sí dentro de un sistema”.

2.6.5 Modelo Relacional

Costa (2002, p.7) define el modelo relacional de la siguiente manera:

¹² MySQL. (s.f.). MySQL: The World's Most Popular Open Source Database. Recuperado en 27 de octubre de 2023, de: https://docs.oracle.com/cd/E17952_01/mysql-8.0-en/what-is-mysql.html.

¹³ Lucidchart. (s.f.). ¿Qué es un diagrama entidad-relación? Lucidchart. Recuperado en 27 de octubre de 2023, de: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>.

El modelo relacional es un modelo de datos y, como tal, tiene en cuenta los tres aspectos siguientes de los datos:

- 1) La estructura, que debe permitir representar la información que nos interesa del mundo real.
- 2) La manipulación, a la que da apoyo mediante las operaciones de actualización y de consulta de los datos.
- 3) La integridad, que es facilitada mediante el establecimiento de reglas de integridad, es decir, condiciones que los datos deben cumplir.

Este tipo de modelo se apoya en sistemas como el de gestión de bases de datos relacional, o en sus siglas SGBDR, y complementa a la definición de datos mediante la estructura de los datos del modelo relacional, así como a la manipulación de estos datos con las operaciones del modelo.

www.bdigital.ula.ve

2.6.6 CRUD

Kroenke y Auer (2015) proporcionan una introducción a los conceptos de bases de datos, incluyendo las operaciones CRUD. Este acrónimo se refiere a las 4 operaciones básicas utilizadas en la gestión de bases de datos:

- **Create (Crear):** Implica la creación de nuevos registros o datos en la base de datos.
- **Read (Leer):** Se refiere a la consulta y a la recuperación de datos existentes en la base de datos.
- **Update (Actualizar):** Involucra la modificación o la actualización de registros o datos existentes en la base de datos.
- **Delete (Eliminar):** Significa la eliminación de registros o datos de la base de datos.

2.7 Metodologías Ágiles

2.7.1 Definición

Según Salesforce ¹⁴, la solución de gestión de relaciones con clientes líder en el mercado, las metodologías ágiles “no son apenas herramientas, sino estrategias integrales (estrategias de marketing, gestión de servicios y más) que impulsan a las organizaciones a gestionar sus proyectos con rapidez y con flexibilidad.” Por otro lado, Cadavid, Martínez y Vélez (2013) señalan que Ghosh (2012) habla de la existencia de los 12 principios de la metodología ágil, siendo creado el *Manifiesto por el desarrollo ágil de software* en 2001, un documento en el que se acuerdan 4 principios básicos para el desarrollo de *software*.

2.7.2 Metodologías Tradicionales vs. Metodologías Ágiles

En la Tabla #3, se comparan las metodologías tradicionales y las metodologías ágiles.

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Tabla #3: Metodologías tradiciones vs. metodologías Ágiles. Fuente: Cadavid, Martínez y Vélez (2013).

¹⁴ Salesforce. (2021, diciembre 22). ¿Qué son las metodologías ágiles? Salesforce Blog. Recuperado en 29 de octubre de 2023, de: <https://www.salesforce.com/mx/blog/que-son-metodologias-agiles/>.

Maida y Pacienza (2015, p.18) comentan que es posible trabajar con cualquiera de las 2 metodologías a la hora de desarrollar un proyecto; también, señalan que:

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta una idea interesante. Estas metodologías pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera, podríamos tener una metodología por cada proyecto, la problemática sería definir cada una de las prácticas, y en el momento preciso definir parámetros para saber cuál usar.

2.7.3 Metodología SCRUM

Cadavid, Martínez y Vélez (2013) comentan que la metodología Scrum para el desarrollo de software es “un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas y de artefactos, y define roles que generan la estructura necesaria para su correcto funcionamiento”. Este tipo de metodología ágil utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos.

Schwaber y Sutherland (2020) mencionan que esta metodología se caracteriza por su enfoque en equipos multidisciplinarios, ciclos de desarrollo cortos llamados "sprints" y la interacción constante con los interesados en el proyecto.

2.7.4 Fases Y Elementos De La Metodología SCRUM

Schwaber y Sutherland (2020, p.3) nombran los requisitos de Scrum. Entre ellos, destaca el Scrum Master para fomentar un entorno donde:

1. Un Product Owner ordena el trabajo de un problema complejo en un Product Backlog.

2. El Scrum Team convierte una selección del trabajo en un Increment de valor durante un Sprint.
3. El Scrum Team y sus interesados inspeccionan los resultados y se adaptan para el próximo Sprint.
4. Repita.

Así mismo, Schwaber y Sutherland (2020, p.7) mencionan un elemento importante dentro de esta metodología, los *Sprints*, en donde:

Las ideas se convierten en valor. Son eventos de duración fija de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior. Todo el trabajo necesario para lograr el Objetivo del Producto, incluido la Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, ocurre dentro de los Sprints.

Para tener una idea ilustrativa de cómo funciona la metodología Scrum, se observa, en la *Figura #2*, la relación entre los principales actores principales, los elementos de planificación y las actividades diarias o finales de un sprint.

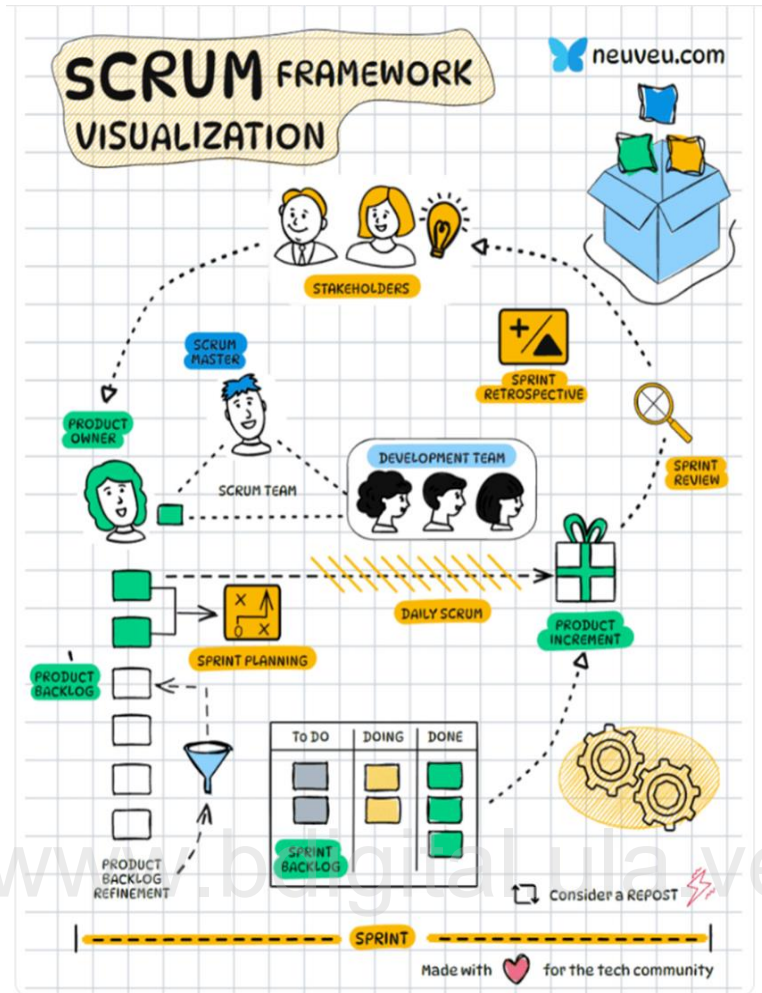


Figura #2: Visualización del framework SCRUM. Fuente: Project Management.

Capítulo III

Marco Metodológico

El marco metodológico es esencial para asegurar la validez y la confiabilidad de los resultados de la investigación. Arias (2012, p. 110) menciona que este marco proporciona una base sólida para la planificación y para la ejecución del estudio, permitiendo a otros investigadores comprender y evaluar la metodología utilizada.

En este capítulo, se describen el tipo y el enfoque de la investigación, las técnicas y los instrumentos para la recolección de datos, así como el procesamiento y la codificación de estos datos. Además, se explica cómo se emplea la metodología de *software* ágil SCRUM para la planificación y para el desarrollo de este proyecto.

3.1 Tipo De Investigación

Este proyecto se enfocó en el desarrollo de una solución tecnológica; específicamente, una aplicación web diseñada para gestionar el registro académico y administrativo de los estudiantes inscritos en algún programa de formación de la SEPEC. El tipo de investigación seleccionado para este proyecto fue el denominado Proyecto Factible. Según Martínez y Vivas (2022), el Proyecto Factible se inscribe en el enfoque cuantitativo, ya que busca analizar e interpretar fenómenos de la realidad con el propósito de encontrar alternativas de solución viables en cualquier contexto educativo (p. 2).

Este proyecto se considera factible y viable porque tuvo como objetivo mejorar la gestión de la información relacionada con los estudiantes, supervisar los pagos de inscripción, los aranceles y el desempeño académico individual. Se buscó lograr consistencia en la información y

optimizar el rendimiento del departamento académico. El proyecto constó de varias etapas, que incluyeron el planteamiento, el diagnóstico, el fundamento teórico, los procedimientos metodológicos y las actividades necesarias para la ejecución y la conclusión del producto.

En este proyecto, se aplicaron conocimientos tecnológicos para abordar los problemas actuales de la SEPEC en la gestión de datos e información del registro académico y administrativo. La optimización se logró mediante el uso de herramientas como el manejo de bases de datos, lo que facilitó el almacenamiento, la actualización, la visualización y la exportación de la información en la solución tecnológica a desarrollar.

3.2 Enfoque De La Investigación

El enfoque de este proyecto es descriptivo, implicando un análisis del problema de la SEPEC presentado en el capítulo I. El objetivo es desarrollar una aplicación web destinada al registro académico y administrativo de los estudiantes, así como al manejo de los datos actuales de la SEPEC. Este análisis se llevó a cabo mediante observación, revisión documental y entrevistas con el propósito de comprender las causas y los efectos relacionados con el desarrollo de la aplicación web.

3.3 Población Y Muestra

3.3.1 Población

Al hablar de la población, se hace referencia al conjunto de personas dentro del ámbito estudiado en el problema investigado. Para Tamayo y Tamayo (2001, p. 30), una población está definida por sus características definitorias; por tanto, el conjunto de elementos que posea esta característica se denomina población o universo. En este proyecto, la población fue la SEPEC.

3.3.2 Muestra

Baravesco (1994) menciona que una muestra es “una parte o fragmento del total de unidades de observación o de análisis sobre los cuales se ha aplicado una selección”. En este proyecto, la muestra fueron los estudiantes y los procesos académicos y administrativos relacionados con la SEPEC, adscrita al Instituto de Investigaciones Económicas y Sociales de la Universidad de Los Andes (ULA), Mérida, Venezuela.

3.3.3 Tipo De Muestra

Para este proyecto, el tipo de muestra es intencional u opinático. Arias (2012) menciona que, en este tipo de muestreo, los elementos “son escogidos con base en criterios o en juicios preestablecidos por el investigador.” (p. 85).

3.4 Instrumento Para La Recolección De Datos

Arias (2012) entiende por técnica de investigación el “procedimiento o forma particular de obtener datos o información.” (p. 67). Yuni y Urbano (2014), como se citó en Mendoza y Avila (2020), mencionan que cualquier herramienta empleada en la recopilación de datos en una investigación científica debe ser precisa, imparcial y poseer validez. Si alguna de estas características no se cumple, la herramienta carecerá de utilidad y los resultados obtenidos no serán válidos.

En este proyecto, la técnica o el instrumento de recolección de datos fueron la observación directa y la entrevista.

3.4.1 Observación Directa

De acuerdo con Hurtado (2008, p. 459), la observación directa se define como un procedimiento que involucra la atención, la recopilación, la selección y el registro de información, en el cual el investigador se apoya en sus sentidos para obtener datos relevantes. En este proyecto, la observación directa proporcionó información de gran importancia para modelar una solución que logre satisfacer las necesidades actuales y los distintos módulos del producto.

3.4.2 Entrevista

Hurtado (2008, p. 469) se refiere a la entrevista como una actividad en la cual 2 individuos se colocan uno frente al otro con el propósito de que una de las personas realice preguntas mientras la otra proporciona respuestas. En este contexto, la entrevista se presenta como un método interactivo de recopilación de información, donde la comunicación directa entre el entrevistador y el entrevistado facilita la obtención de datos detallados y específicos sobre el tema en cuestión.

3.5 Procesamiento Y Codificación De Datos

Los registros de notas de las observaciones y las entrevistas de la SEPEC llevaron a obtener información clara y precisa sobre la situación de la misma, lo que determinó el diagnóstico previo del problema presentado. Para el desarrollo de este proyecto, se cumplieron las siguientes fases:

- **Fase N.º 1 - Etapa Documental:** Recopilación de material bibliográfico referente al producto a desarrollar y entregar, lo que permitió entender como otros proyectos han efectuado aplicaciones web para el registro académico y administrativo.

- **Fase N.º 2 - Etapa de Obtención de Datos:** Obtención de los datos necesarios, cumpliendo con la actividad de asistir a las instalaciones de la SEPEC para modelar los datos a registrar.
- **Fase N.º 3 - Etapa Analítica:** Se establecieron los criterios de evaluación y de análisis de la información respecto a A) requerimientos y necesidades con respecto a la base de datos para una unidad de estudios de posgrado, B) diseño de los diferentes módulos y de la interfaz gráfica del producto, C) establecer la formulación del proyecto y el desarrollo del mismo, y D) elaborar las conclusiones y las recomendaciones de acuerdo a los resultados obtenidos de este proyecto.
- **Fase N.º 4 – Etapa de Ejecución:** Se establecieron los sprints siguiendo la metodología SCRUM para el desarrollo del proyecto; además, se incluyeron las mejoras y se trabajó bajo los requerimientos del proyecto y de las correcciones encontradas en las reuniones realizadas con el tutor.
- **Fase N.º 5 – Etapa Final:** Se realizó la redacción y la entrega del manuscrito final, que incluye las recomendaciones y las conclusiones a las que se llegaron tras completar el desarrollo de la aplicación web.

3.6 Cronograma De Actividades

A continuación, se describe el conjunto de actividades que permitieron el desarrollo del presente proyecto de grado. La fecha de inicio fue el lunes 22 de enero de 2024 y la fecha de finalización el viernes 31 de mayo de 2024, excluyendo los días feriados. La Tabla #4 muestra el

cronograma de actividades para la elaboración del proyecto.

Fase	Actividad	Semana															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Fase 1	Actividad 1																
Fase 2	Actividad 2																
Fase 3	Actividad 3																
Fase 4	Actividad 4																
	Actividad 5																
Fase 5	Actividad 6																
	Actividad 7																

Tabla #4. Cronograma de Actividades. Fuente: Propia.

3.7 Cronograma De Evaluaciones

En la Tabla #5, se muestra el cronograma de evaluaciones implementado para el avance y el desarrollo del presente proyecto de grado.

Evaluación	Semana															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Reuniones con el tutor																
Presentación de avance del proyecto de grado																
Entrega preliminares																
Entrega del manuscrito final																
Defensa del proyecto de grado																

Tabla #5. Cronograma de evaluaciones. Fuente: Propia.

3.8 Aplicación De La Metodología SCRUM

Se considera factible la metodología ágil SCRUM para aplicarla en el desarrollo de la aplicación web para la SEPEC, ya que se adapta al desarrollo del proyecto.

3.9 Desarrollo De La Metodología SCRUM

Cada uno de los entregables tuvo una duración de 1 a 3 semanas, que se conocen como *Sprint* en los que se desarrollaron diferentes fases del proyecto. En el capítulo 4, se presenta el cronograma del Sprint Backlog junto a las historias de usuario, a los requerimientos, al tipo de prioridad de una historia, al tiempo de ejecución, entre otros.

3.9.1 Alcance

El proyecto a desarrollar y entregar debe cumplir con estos objetivos:

- Tendrá un apartado de inicio en donde el usuario podrá iniciar sesión ingresando su usuario y contraseña.
- Se podrá realizar el CRUD (create, read, update, delete) para todos los módulos desarrollados, ya sea inscripción, solicitud, entre otros.
- Se podrán registrar las calificaciones obtenidas de las asignaturas inscritas en un semestre por cada estudiante.
- Deberá facilitar la obtención de información relacionada con contacto de un estudiante, con matriculación y con asignaturas inscritas.
- Generará facturas por gastos de aranceles, por cursos o por asignaturas inscritas por cada estudiante.
- Incluirá una sección de pagos para llevar un control administrativo de los pagos registrados en un mes.
- Se podrá descargar archivos con extensión o formato PDF o Excel al momento de generar algún reporte académico o administrativo.

3.9.2 Roles

- **Scrum Master:** Encargado de que se emplee correctamente la metodología según las guías de Scrum. Para efectos de este proyecto, el encargado será el Br. Abraham Mousalli.
- **Product Owner:** Encargado de hacer las reuniones y de definir las prioridades. En este caso, es el director de la SEPEC y la secretaria de asuntos estudiantiles.
- **Development Team:** Es el equipo o persona encargada del desarrollo del producto. La persona encargada será el Br. Abraham Mousalli.

www.bdigital.ula.ve

Capítulo IV

Extracción De Requerimientos

En este capítulo, se recopilan los requerimientos para el desarrollo de la aplicación web. Se trabajó bajo un formato para tener la misma consistencia de cada uno de los requisitos, aplicando los principios de la metodología SCRUM para, así, crear un Product Backlog dividido en sprint; cada sprint está acompañado de las historias de usuario a desarrollar.

4.1 Formato De Los Requerimientos

4.1.1 Historias De Usuario

Las historias de usuario describen, de manera breve y sencilla, una funcionalidad o característica deseada desde la perspectiva de un usuario particular o de un cliente; además, tiene una estructura típica, para este proyecto, se utilizará el formato tradicional cómo (rol), quiero (objetivo) y para (beneficio).

4.1.2 Usuario

Rol dentro de la aplicación web o sistema que requiere de una funcionalidad en concreto.

4.1.3 Priorización

La priorización indica la importancia relativa de cada historia de usuario en comparación con el resto de historias contempladas dentro de un Product Backlog. A fines de priorizar las historias de este proyecto, se etiquetarán como Alta, Media o Baja.

4.1.4 Criterios De Aceptación

Son condiciones específicas que deben cumplirse para que una historia de usuario se pueda considerar como completa y aceptable. A fines de este proyecto, los criterios de aceptación ayudarán a entender exactamente qué se necesita desarrollar y cómo se evalúa la funcionalidad.

4.1.5 Requisitos Funcionales

Describen lo que el sistema debe hacer e incluyen desde características y funcionalidades hasta comportamientos para lograr el funcionamiento de cada historia de usuario.

4.1.6 Limitaciones

Pueden ser técnicas, legales, de tiempo, recursos o cualquier otra restricción que el equipo debe considerar y establecen restricciones o condiciones que pueden afectar la implementación de una historia de usuario.

4.1.7 Dependencia

Permite relacionar las dependencias entre las historias de usuario.

4.1.8 Comentario

Ayuda a complementar la historia de usuario, agregando información importante o que se debe considerar a la hora de trabajar en dicha historia.

4.2 Lista De Requerimientos

4.2.1 Configuración Del Proyecto En Django

- **Historia de usuario #1.**
- **Usuario:** Desarrollador.
- **Descripción:** Como desarrollador quiero configurar el proyecto inicial en Django para establecer la base y la estructura del proyecto de manera organizada.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Crear un nuevo proyecto de Django usando el comando `django-admin startproject`.
 - Establecer la configuración inicial en el archivo `settings.py`; por ejemplo, base de datos, zonas horarias, idiomas, etc.
 - Crear un entorno virtual para el proyecto.
 - Instalar las dependencias necesarias; por ejemplo, Django, etc.
 - Configurar los archivos `urls.py` y `wsgi.py` correctamente.
- **Requerimientos funcionales:**
 - Creación del proyecto Django.
 - Configuración inicial del archivo `settings.py`.
 - Configuración del entorno virtual.
 - Instalación de dependencias.
 - Configuración de archivos `urls.py` y `wsgi.py`.
- **Limitaciones:** Ninguna.
- **Dependencia:** Ninguna.

- **Comentario:** Ninguno.

4.2.2 Crear Aplicaciones Modulares En El Proyecto Django

- **Historia de usuario #2.**
- **Usuario:** Desarrollador.
- **Descripción:** Como desarrollador, quiero crear aplicaciones modulares dentro del proyecto Django para mantener una estructura ordenada que permita facilitar la gestión y el desarrollo del proyecto.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Permitir la creación de *apps* individuales usando el comando `python manage.py startapp`.
 - Incluir *apps* básicas como *students*, *products*, *courses*, *bills*, *records*, y *reports*.
 - Cada *app* debe tener sus propios modelos, vistas, urls y templates organizados.
 - Las *apps* deben ser registradas en el archivo `settings.py` del proyecto principal.
 - Cada *app* debe incluir un archivo `admin.py` para la administración en el panel de Django.
- **Requerimientos funcionales:**
 - Creación de las *apps* individuales.
 - Organización de modelos, de vistas, de urls y de templates dentro de cada *app*.
 - Registro de las *apps* en `settings.py`.
 - Inclusión de archivo `admin.py` en cada *app*.
- **Limitaciones:** Ninguna.
- **Dependencia:** #1.

- **Comentario:** Ninguno.

4.2.3 Establecer La Arquitectura De Ficheros Del Proyecto Django

- **Historia de usuario #3.**
- **Usuario:** Desarrollador.
- **Descripción:** Como desarrollador, quiero definir una arquitectura de ficheros clara y organizada para facilitar la navegación y el mantenimiento del proyecto.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Tener un directorio raíz para el proyecto con un nombre significativo.
 - Tener directorios separados para cada *app*; por ejemplo, students, courses, bills y reports.
 - Incluir un directorio general templates para las plantillas HTML.
 - Incluir un directorio general static para archivos CSS, JS e imágenes.
 - Incluir un archivo requirements.txt para las dependencias del proyecto.
- **Requerimientos funcionales:**
 - Organización de directorios y de ficheros.
 - Inclusión de directorios templates con carpetas de cada *app* y un static general.
 - Configuración de archivo requirements.txt.
- **Limitaciones:** Ninguna.
- **Dependencia:** #1.
- **Comentario:** Seguir buenas prácticas de estructura de proyectos en Django y usar un archivo .env para variables de entorno.

4.2.4 Configurar El Archivo settings.py Del Proyecto Django

- **Historia de usuario #4.**
- **Usuario:** Desarrollador.
- **Descripción:** Como desarrollador, quiero configurar adecuadamente el archivo settings.py, para asegurar el correcto funcionamiento del proyecto con todas las configuraciones necesarias.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe configurar las bases de datos en DATABASES.
 - Debe configurar las zonas horarias y el idioma en TIME_ZONE y LANGUAGE_CODE.
 - Debe configurar las rutas de archivos estáticos y de media en STATIC_URL y MEDIA_URL.
 - Debe configurar las aplicaciones instaladas en INSTALLED_APPS.
 - Debe incluir configuraciones de seguridad (por ejemplo, SECRET_KEY, DEBUG, ALLOWED_HOSTS).
 - Debe configurar los middleware en MIDDLEWARE.
- **Requerimientos funcionales:**
 - Configuración de base de datos.
 - Configuración de zonas horarias y de idioma.
 - Configuración de archivos estáticos y de media.
 - Registro de aplicaciones instaladas.
 - Configuración de seguridad.

- Configuración de *middleware*.
- **Limitaciones:** Ninguna.
- **Dependencia:** #1.
- **Comentario:** Ninguno.

4.2.5 Configuración Inicial De La Base De Datos

- **Historia de usuario #5.**
- **Usuario:** Desarrollador.
- Descripción: Como desarrollador, quiero configurar la base de datos inicial para tener una estructura sólida y funcional para el resto del desarrollo.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La base de datos debe estar configurada y ser accesible para MySQL.
 - Las tablas iniciales (Students, Phone, StudentPhone, Email, StudentEmail, Nationality, StudentNationality) deben estar creadas con sus respectivas relaciones.
 - Se deben establecer migraciones iniciales para generar las primeras tablas de la base de datos.
- **Requerimientos funcionales:**
 - Configuración de la conexión a la base de datos en el entorno de desarrollo.
 - Creación de modelos iniciales y de sus relaciones.
 - Migraciones aplicadas correctamente y tablas generadas en MySQL.
- **Limitaciones:** Tener, previamente, el proyecto en Django y el acceso a la base de datos previamente creada.

- **Dependencia:** #2 y #3.
- **Comentario:** Ninguno.

4.2.6 Sistema De Login

- **Historia de usuario #6.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder iniciar sesión en la aplicación para acceder al sistema y para utilizar los diferentes módulos.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El usuario debe poder acceder a una página de inicio de sesión.
 - El sistema debe autenticar al usuario utilizando su nombre de usuario y su contraseña.
 - El sistema debe proporcionar retroalimentación en caso de errores; por ejemplo, nombre de usuario o contraseña incorrectos.
 - Después de iniciar sesión, el usuario debe ser redirigido a la página principal.
 - El sistema debe manejar sesiones de usuario de manera segura.
- **Requerimientos funcionales:**
 - **Página de Inicio de Sesión:** Crear una vista y una plantilla sencilla para el formulario de inicio de sesión.
 - **Autenticación:** Implementar la lógica de autenticación utilizando el sistema de autenticación de Django (authenticate, login).
 - **Manejo de Errores:** Mostrar mensajes de error claros y específicos si el inicio de sesión falla.

- Redirección: Redirigir al usuario a la página correspondiente después de iniciar sesión.
- **Limitaciones:** Debe garantizarse la seguridad de las credenciales del usuario.
- **Dependencia:** #5.
- **Comentario:** Ninguno.

4.2.7 Crear Un Nuevo Estudiante

- **Historia de usuario #7.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar nuevos estudiantes en la base de datos para mantener un registro completo y actualizado de todos los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El formulario de registro debe incluir todos los campos obligatorios: identificación del estudiante (C.I. o Pasaporte), nombre, apellido, fecha de nacimiento y profesión.
 - La identificación del estudiante (C.I. o Pasaporte) debe ser única y validada.
 - El sistema debe mostrar mensajes de error específicos si hay campos faltantes o datos no válidos.
 - Después de registrar al estudiante, el sistema debe mostrar una confirmación de éxito y redirigir a la lista de estudiantes.
- **Requerimientos funcionales:**

- Formulario de Registro: Crear un formulario para la entrada de datos del estudiante.
- Validación de Datos: Implementar validaciones para la identificación del estudiante y otros campos obligatorios.
- Confirmación y Redirección: Mostrar mensaje de éxito y redirigir a la lista de estudiantes.
- **Limitaciones:** La identificación del estudiante debe ser única y no debe permitir duplicados.
- **Dependencia:** #2.
- **Comentario:** Incluir la unión de tablas como StudentNationality, StudentEmail y StudentPhone en el formulario.

4.2.8 Actualizar Información Del Estudiante

- **Historia de usuario #8.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la información de los estudiantes para mantener datos actualizados y correctos.
- **Prioridad:** Media.
- **Criterios de aceptación:**
 - El sistema debe permitir la edición de todos los campos del estudiante.
 - Debe validarse la identificación del estudiante para evitar duplicados.
 - Al guardar los cambios, debe mostrarse una confirmación de éxito.
 - La fecha de actualización debe ser registrada automáticamente.
- **Requerimientos funcionales:**

- Formulario de Edición: Crear una vista utilizando el mismo formulario para editar los datos del estudiante.
- Validación de Datos: Implementar validaciones para asegurar que la identificación del estudiante sigue siendo única.
- Registro de Fecha de Actualización: Actualizar automáticamente el campo `update_day`.
- **Limitaciones:** Debe permitirse la edición del campo C.I./Pasaporte siempre y cuando no cause duplicados.
- **Dependencia:** #7.
- **Comentario:** Incluir, en la actualización de los datos, los campos provenientes de `StudentNationality`, `StudentEmail` y `StudentPhone`.

4.2.9 Eliminar Estudiante

- **Historia de usuario #9.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder eliminar registros de estudiantes para mantener la base de datos limpia y libre de registros obsoletos o incorrectos.
- **Prioridad:** Baja.
- **Criterios de aceptación:**
 - El sistema debe pedir confirmación antes de eliminar un registro.
 - Si un estudiante tiene datos relacionados (por ejemplo, facturas, matriculaciones o inscripciones), el sistema debe manejar estas relaciones adecuadamente y evitar su eliminación.

- Después de eliminar un estudiante, debe mostrarse una confirmación de éxito.
- **Requerimientos funcionales:**
 - Confirmación de Eliminación: Mostrar un cuadro de diálogo para confirmar la eliminación.
 - Manejo de Relaciones: Asegurar que los datos relacionados se manejen correctamente (eliminar en cascada o mantener integridad referencial).
 - Confirmación de Éxito: Mostrar mensaje de éxito después de la eliminación.
- **Limitaciones:** No debe eliminarse el usuario si tiene relación con otras tablas como, por ejemplo, StudentRegistration.
- **Dependencia:** #7.
- **Comentario:** Inhabilitar el botón de eliminar si tiene ya una relación con otra tabla.

4.2.10 Visualizar Estudiante

- **Historia de usuario #10.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder ver la información detallada de cada estudiante para acceder rápidamente a los datos necesarios.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El sistema debe mostrar todos los campos de información del estudiante.
 - La visualización debe ser clara y organizada.
 - Debe ser posible acceder fácilmente a la información de los datos del estudiante como correos, números de teléfono, nacionalidad o datos personales.
- **Requerimientos funcionales:**

- Vista Detallada: Crear una página de detalle para mostrar la información del estudiante.
- Acceso a Datos Relacionados: Incluir enlaces o secciones para visualizar números de teléfono, correos electrónicos y nacionalidad.

Limitaciones: Ninguna.

Dependencia: #7.

Comentario: Incluir un botón para poder actualizar la información.

4.2.11 Crear Una Carrera Nueva

- **Historia de usuario #11.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder crear una nueva carrera con nombre, con unidades de crédito y con duración para ofrecer nuevos programas educativos a los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La carrera debe tener un nombre único.
 - Las unidades de crédito deben estar entre 1 y 100.
 - La duración debe ser seleccionada dentro de las opciones desplegadas como 2 años, 5 años, entre otros.
- **Requerimientos funcionales:**
 - Interfaz de usuario para ingresar los datos de la carrera.
 - Validaciones para los campos de nombre, de unidades de crédito y de duración.
- **Limitaciones:** Ninguna.

- **Dependencia:** #2.
- **Comentario:** Considerar la posibilidad de cambios futuros en las opciones de duración.

4.2.12 Actualizar Una Carrera

- **Historia de usuario #12.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la información de una carrera existente para mantener la información del programa actualizada y precisa.
- **Prioridad:** Media.
- **Criterios de aceptación:**
 - La actualización debe permitir cambios en el nombre, en las unidades de crédito y en la duración.
 - Se debe registrar la fecha de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para modificar los datos de una carrera existente.
 - Validaciones para asegurar que los cambios sean coherentes.
- **Limitaciones:** Ninguna.
- **Dependencia:** #11.
- **Comentario:** Redireccionar a la vista de la carrera al momento de actualizar.

4.2.13 Crear Una Cohorte

- **Historia de usuario #13.**
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder crear una nueva cohorte asociada a una carrera para organizar a los estudiantes en cohortes específicas según su año de inicio.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La cohorte debe tener un nombre único dentro de la misma carrera.
 - La cohorte debe estar asociada a una carrera específica.
 - La fecha de inicio debe ser especificada al momento de crearla.
- **Requerimientos funcionales:**
 - Interfaz de usuario para ingresar los datos de la cohorte.
 - Validaciones para asegurar la unicidad del nombre dentro de la carrera.
- **Limitaciones:** Haber creado la carrera con anterioridad.
- **Dependencia:** #2.
- **Comentario:** Ninguno.

4.2.14 Calcular Fecha Posible De Culminación

- **Historia de usuario #14.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero que el sistema calcule, automáticamente, la fecha posible de culminación de una cohorte basada en la duración de la carrera para facilitar la planificación y el seguimiento del progreso de los estudiantes.
- **Prioridad:** Media.
- **Criterios de aceptación:**

- La fecha posible de culminación debe calcularse automáticamente al crear la cohorte.
- La fecha posible de culminación debe ser editable posteriormente.
- **Requerimientos funcionales:**
 - Lógica de negocio para calcular la fecha posible de culminación.
 - Interfaz de usuario para mostrar y para editar la fecha de culminación junto con la cohorte.
- **Limitaciones:** Ninguna.
- **Dependencia:** #13
- **Comentario:** Utilizar la duración de la carrera en años para calcular la fecha tentativa de culminación de un estudiante.

4.2.15 Definir Un Semestre Actual

- **Historia de usuario #15.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder definir y activar un semestre actual para gestionar el período académico en curso.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Solo un semestre puede estar activo a la vez.
 - El semestre debe tener un nombre único.
- **Requerimientos funcionales:**
 - Interfaz de usuario para activar un semestre.
- **Limitaciones:** Ninguna.

- **Dependencia:** #2.
- **Comentario:** Ninguno.

4.2.16 Actualizar Fechas Del Semestre

- **Historia de usuario** #16.
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar las fechas de inicio y de culminación del semestre para reflejar cambios en el calendario académico.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Las fechas de inicio y de culminación deben ser editables.
 - Las actualizaciones deben registrarse con una fecha de modificación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para editar las fechas del semestre.
- **Limitaciones:** Ninguna.
- **Dependencia:** #15.
- **Comentario:** La fecha de inicio y de culminación serán utilizadas luego para generar el reporte de calificación del estudiante.

4.2.17 Registrar Una Asignatura

- **Historia de usuario** #17.
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder registrar una nueva materia con su código, con su nombre y con sus unidades de crédito para ofrecer una variedad de asignaturas a los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El código de la materia debe ser único.
 - Las unidades de crédito deben estar entre 1 y 20.
 - La materia debe estar asociada, obligatoriamente, a una carrera.
- **Requerimientos funcionales:**
 - Interfaz de usuario para ingresar los datos de la materia.
 - Validaciones para asegurar la unicidad del código y el rango de las unidades de crédito.
- **Limitaciones:** Ninguna.
- **Dependencia:** #2.
- **Comentario:** Permitir la edición de materias existentes y no poder eliminarlas si ya están incluidas en un StudyPlan.

4.2.18 Asociar Una Asignatura A Una Carrera

- **Historia de usuario #18.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero asociar una asignatura específica a una carrera para asegurar que las asignaturas ofrecidas correspondan al mismo plan de estudio seleccionado.
- **Prioridad:** Alta.

- **Criterios de aceptación:**
 - La asignatura debe estar asociada a una carrera específica.
 - Las asociaciones deben ser editables.
 - Impedir eliminar la asociación si ya un estudiante ha inscrito la asignatura.
- **Requerimientos funcionales:**
 - Interfaz de usuario para asociar asignaturas a planes de estudio.
 - Validaciones para asegurar que la materia pertenezca a la carrera correcta.
- **Limitaciones:** haber creado la carrera con anterioridad.
- **Dependencia:** #11 y #17.
- **Comentario:** Permitir la visualización de todas las asignaturas asociadas a un plan de estudio.

4.2.19 Crear Un Plan De Estudio

- **Historia de usuario #19.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder crear un nuevo plan de estudios con su nombre, con su descripción y con una carrera asociada para estructurar el plan académico que deben seguir los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El nombre del plan de estudio debe ser único.
 - El plan de estudio debe estar asociado a una carrera específica.
 - La descripción es opcional.
- **Requerimientos funcionales:**

- Interfaz de usuario para asociar asignaturas a planes de estudio.
- Validaciones para asegurar que la materia pertenezca a la carrera correcta.
- **Limitaciones:** Ninguna.
- **Dependencia:** #11.
- **Comentario:** Ninguno.

4.2.20 Actualizar Un Plan De Estudio

- **Historia de usuario #20.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la información de un plan de estudio existente para reflejar cambios y mejoras en la estructura educativa.
- **Prioridad:** Media.
- **Criterios de aceptación:**
 - La actualización debe permitir cambios en el nombre, en la descripción y en la carrera asociada.
 - Se debe registrar la fecha de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para modificar los datos de un plan de estudios existente.
 - Validaciones para asegurar que los cambios sean coherentes.
- **Limitaciones:** Ninguna.
- **Dependencia:** #19.
- **Comentario:** Ninguno.

4.2.21 Asignar Una Asignatura A Un Semestre Y A Un Plan De Estudio

- **Historia de usuario #21.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder asignar una asignatura a un semestre específico dentro de un plan de estudios para organizar la secuencia de las asignaturas a lo largo del programa académico.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La asignatura debe estar asociada a un semestre específico.
 - La asignación debe ser editada y actualizada correctamente.
- **Requerimientos funcionales:**
 - Interfaz de usuario para asignar materias a semestres.
 - Validaciones para asegurar la correcta asociación de materias y de semestres.
- **Limitaciones:** Ninguna.
- **Dependencia:** #17 y #19.
- **Comentario:** Permitir la visualización de todas las asignaturas asociadas a un plan de estudio.

4.2.22 Actualizar Asignatura Asociada A Un Plan De Estudio

- **Historia de usuario #22.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la asignación de una materia a un semestre para reflejar cambios en la estructura del plan de estudios.

- **Prioridad:** Baja.
- **Criterios de aceptación:**
 - La asignatura solo podrá actualizarse en la etapa del semestre en la que se encuentra.
- **Requerimientos funcionales:**
 - Interfaz de usuario para actualizar la etapa del semestre de la asignatura respecto al plan de estudio.
- **Limitaciones:** Ninguna.
- **Dependencia:** #21.
- **Comentario:** Ninguno.

4.2.23 Matricular Un Estudiante

- **Historia de usuario #23.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder matricular a un estudiante en un plan de estudios y en una cohorte específica para registrar, oficialmente, su participación en un programa académico.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El estudiante debe estar asociado a un plan de estudios y a una cohorte específicos.
 - La matriculación debe registrar la fecha de matriculación y de actualización si hay futuros cambios.
- **Requerimientos funcionales:**

- Interfaz de usuario para matricular estudiantes.
- Validaciones para asegurar la correcta asociación del estudiante al plan de estudios y a la cohorte.
- **Limitaciones:** Tener un registro en el sistema del estudiante a matricular.
- **Dependencia:** #7, #15 y #18.
- **Comentario:** Permitir la edición y la actualización de la matriculación del estudiante.

4.2.24 Actualizar El Estatus Del Estudiante En Su Matriculación

- **Historia de usuario #24.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar el estatus académico de un estudiante (activo, completado, retirado, etc.) para mantener un seguimiento preciso de su progreso y de su situación académica.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - El estatus del estudiante debe ser editable.
 - Se debe registrar la fecha de actualización del estatus.
- **Requerimientos funcionales:**
 - Interfaz de usuario para actualizar el estatus del estudiante en la matriculación de un plan de estudio.
 - Validaciones para asegurar que el cambio de estatus sea coherente.
 - **Limitaciones:** Ninguna.
 - **Dependencia:** #23.

- **Comentario:** El único estatus en el que se podrá inscribir el estudiante es si está como Activo.

4.2.25 Registrar La Inscripción De Un Estudiante En Una Materia

- **Historia de usuario #25.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder inscribir a un estudiante en una asignatura específica dentro de su semestre activo y su plan de estudios para asegurar que los estudiantes se matriculen correctamente en las asignaturas correspondientes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La inscripción debe asociar al estudiante con el plan de estudios, con el semestre activo y con la asignatura específica.
 - La inscripción debe registrar la fecha de creación.
 - Debe garantizarse la unicidad de la combinación de estudiante, de semestre y de materia.
 - El listado de asignaturas a inscribir debe mostrar solo las asignaturas faltantes por cursar. Debe excluir cualquier asignatura ya cursada, aprobada o por equivalencia.
- **Requerimientos funcionales:**
 - Interfaz de usuario para inscribir a un estudiante en una asignatura.
 - Validaciones para asegurar la correcta asociación y la unicidad de la inscripción.
- **Limitaciones:** Ninguna.
- **Dependencia:** #7, #15 y #18.

- **Comentario:** La inscripción debe reflejarse inmediatamente en el historial académico del estudiante.

4.2.26 Actualizar La Inscripción De Un Estudiante

- **Historia de usuario #26.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la inscripción de un estudiante en una asignatura para reflejar cambios necesarios en la matrícula del estudiante.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La actualización debe permitir cambios en el semestre y en la materia.
 - Se debe registrar la fecha de actualización.
 - El listado de asignaturas a inscribir debe mostrar solo las asignaturas faltantes por cursar. Debe excluir cualquier asignatura ya cursada, aprobada o por equivalencia.
- **Requerimientos funcionales:**
 - Interfaz de usuario para modificar la inscripción de un estudiante.
 - Validaciones para asegurar que los cambios sean coherentes y se mantenga la unicidad.
- **Limitaciones:** Si ya la asignatura fue registrada en una factura, no se podrá actualizar.
- **Dependencia:** #25.
- **Comentario:** Ninguno.

4.2.27 Visualizar Todas Las Inscripciones De Un Estudiante

- **Historia de usuario #27.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder visualizar todas las inscripciones de un estudiante en diferentes semestres y asignaturas para tener un seguimiento completo del progreso académico del estudiante.
- **Prioridad:** Media.
- **Criterios de aceptación:**
 - La vista debe mostrar todas las inscripciones con detalles de plan de estudios, de semestre y de materia.
 - Debe incluir fechas de creación y de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para visualizar todas las inscripciones de un estudiante.
 - **Limitaciones:** Ninguna.
 - **Dependencia:** #25.
 - **Comentario:** Ninguno.

4.2.28 Registrar Calificación De Una Asignatura

- **Historia de usuario #28.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar la calificación de una asignatura inscrita por un estudiante para evaluar el desempeño académico del estudiante.

- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La calificación debe asociarse a la inscripción de la asignatura específica del estudiante.
 - Debe incluir el estado de la asignatura (aprobado, reprobado, etc.).
 - La fecha de creación debe ser registrada automáticamente.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar calificaciones.
 - Validaciones para asegurar la correcta asociación de calificaciones con inscripciones.
- **Limitaciones:** Ninguna.
- **Dependencia:** #25.
- **Comentario:** Permitir la inclusión de observaciones adicionales sobre el rendimiento del estudiante.

4.2.29 Actualizar Una Calificación Existente

- **Historia de usuario #29.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar la calificación y el estado de una materia inscrita por un estudiante para reflejar correcciones o revisiones en el rendimiento del estudiante.
- **Prioridad:** Media.
- **Criterios de aceptación:**

- La actualización debe permitir cambios en la calificación y en el estado de la asignatura.
- Se debe registrar automáticamente la fecha de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para modificar calificaciones y estados de materias.
 - Validaciones para asegurar que los cambios sean coherentes.
- **Limitaciones:** Si el semestre en que se inscribió no se encuentra activo, no se podrá actualizar ni eliminar la calificación.
- **Dependencia:** #28.
- **Comentario:** Ninguno.

4.2.30 Visualizar El Historial De Calificaciones De Un Estudiante

- **Historia de usuario #30.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder visualizar el historial completo de calificaciones y los estados de asignaturas de un estudiante para tener un seguimiento detallado de su rendimiento académico a lo largo del tiempo.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - La vista debe mostrar todas las calificaciones y los estados con detalles de inscripción de asignaturas.
 - Debe incluir fechas de creación y de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para visualizar el historial de calificaciones de un estudiante.

- **Limitaciones:** Ninguna.
- **Dependencia:** #7 y #28.
- **Comentario:** Permitir la exportación de esta información para informes académicos e incluir el ciclo de duración del semestre, datos del estudiante y calificaciones cursadas y/o registradas.

4.2.31 Registrar Un Nuevo Arancel, Un Curso O Una Unidad Curricular

- **Historia de usuario #31.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar un nuevo arancel, curso o unidad curricular con su nombre y con su descripción para gestionar y categorizar aranceles, cursos o unidades curriculares aplicables a diferentes servicios que ofrece la institución.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la creación de un arancel, de curso o de una unidad curricular con nombre y con descripción.
 - El nombre del arancel, del curso o de la unidad curricular debe ser único.
 - Se necesita un buscador para facilitar encontrar un arancel, un curso o una unidad curricular.
- **Requerimientos funcionales:**
 - Interfaces de usuario independientes para registrar nuevos aranceles, cursos o unidades curriculares.
 - Validaciones para asegurar la unicidad del nombre.

- Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Ninguna.
- **Dependencia:** #2.
- **Comentario:** La descripción del arancel, del curso o de la unidad curricular es opcional.

4.2.32 Actualizar Un Nuevo Arancel, Un Curso O Una Unidad Curricular

- **Historia de usuario #32.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar el campo de nombre y la descripción de un arancel, de un curso o de una unidad curricular para gestionar y para categorizar los aranceles, los cursos y las unidades curriculares que se incluirán como productos en la facturación.
- **Prioridad:** Baja.
- **Criterios de aceptación:**
 - Debe permitir la actualización de la descripción sin cambiar el nombre.
- **Requerimientos funcionales:**
 - Interfaz de usuario para modificar la descripción de un arancel, de un curso o de una unidad curricular.
 - Notificación de éxito o de error al momento de actualizar.
- **Limitaciones:** Ninguna.
- **Dependencia:** #31.
- **Comentario:** Ninguno.

4.2.33 Registrar El Precio De Un Arancel Para Una Nacionalidad Específica

- **Historia de usuario #33.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar el precio de un arancel específico para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la asociación de un arancel con una nacionalidad y con su precio.
 - Debe registrar, automáticamente, la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar precios de aranceles.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.
- **Dependencia:** #31.
- **Comentario:** Ninguno.

4.2.34 Actualizar El Precio De Un Arancel Para Una Nacionalidad Específica

- **Historia de usuario #34.**
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder actualizar el precio de un arancel específico para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la asociación de un arancel con una nacionalidad y con su precio.
 - Debe registrar, automáticamente, la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para actualizar precios de aranceles.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de actualizar.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.
- **Dependencia:** #33.
- **Comentario:** Ninguno.

4.2.35 Registrar El Precio De Un Curso Para Una Nacionalidad Específica

- **Historia de usuario #35.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar el precio de un curso para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la asociación de un curso con una nacionalidad y con su precio.

- Debe registrar, automáticamente, la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar precios de aranceles.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.
- **Dependencia:** #31.
- **Comentario:** Ninguno.

4.2.36 Actualizar El Precio De Un Curso Para Una Nacionalidad Específica

- **Historia de usuario #36.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar el precio de un curso para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta
- **Criterios de aceptación:**
 - Debe permitir la asociación de un curso con una nacionalidad y con su precio.
 - Debe registrar, automáticamente, la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para actualizar precios de cursos.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de actualizar.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.

- **Dependencia:** #35.
- **Comentario:** Ninguno.

4.2.37 Registrar El Precio De Una Unidad Curricular Para Una Nacionalidad Específica

- **Historia de usuario #37.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar el precio de una unidad curricular para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta
- **Criterios de aceptación:**
 - Debe permitir la asociación de una unidad curricular con una nacionalidad y con su precio.
 - Debe registrar la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar precios de unidades curriculares.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.
- **Dependencia:** #31.
- **Comentario:** Ninguno.

4.2.38 Actualizar El Precio De Una Unidad Curricular Para Una Nacionalidad Específica

- **Historia de usuario #38.**

- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder actualizar el precio de una unidad curricular específica para una nacionalidad con el fin de gestionar los precios basados en la nacionalidad de los estudiantes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la asociación de una unidad curricular con una nacionalidad y con su precio.
 - Debe registrar la fecha **de creación**.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar precios de unidades curriculares.
 - Validaciones para asegurar la correcta asociación.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Solo un precio por tipo de nacionalidad creada.
- **Dependencia:** #37.
- **Comentario:** Ninguno.

4.2.39 Crear Una Nueva Factura Para Un Estudiante

- **Historia de usuario #39.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder crear una nueva factura para un estudiante con el fin de registrar y de gestionar los cobros pendientes.
- **Prioridad:** Alta.
- **Criterios de aceptación:**

- Debe permitir el ingreso de una fecha de vencimiento.
- El sistema debe permitir la selección del estado de la factura (Pendiente o Cancelada).
- El sistema debe permitir la selección de la condición de pago (Contado o Crédito).
- El sistema debe calcular y mostrar el subtotal, el descuento, el monto total a pagar y el saldo del estudiante.
- El sistema debe registrar, automáticamente, la fecha de creación y de actualización.
- **Requerimientos funcionales:**
 - Interfaz de usuario para crear nuevas facturas.
 - Generación automática de números de factura únicos.
 - Validación de datos de entrada.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Inicialmente, la factura no tiene ningún producto asociado.
- **Dependencia:** #2 y #7.
- **Comentario:** La factura utilizará la nacionalidad del estudiante para mostrar los precios de productos, de aranceles, de cursos y de unidades curriculares de acuerdo a su nacionalidad.

4.2.40 Agregar Un Elemento A Una Factura Existente

- **Historia de usuario #40.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder agregar un elemento a una factura existente para detallar los cargos específicos que el estudiante debe pagar.

- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Utilizar campos de selección de producto para mostrar, dinámicamente, los tipos de productos asociados y los precios de acuerdo al tipo de nacionalidad del estudiante.
 - El sistema debe calcular y mostrar el total del producto, considerando posibles descuentos.
 - El sistema debe registrar, automáticamente, la fecha de creación.
- **Requerimientos funcionales:**
 - Interfaz de usuario para agregar elementos a una factura.
 - Validación de datos de entrada.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Si el estudiante no tiene asignaturas inscritas, no se podrá incluir a la factura productos del tipo unidad curricular.
- **Dependencia:** #33, #34, #37 y #39.
- **Comentario:** La factura utilizará la nacionalidad del estudiante para mostrar los precios de productos, de aranceles, de cursos y de unidades curriculares de acuerdo a su nacionalidad.

4.2.41 Registrar Un Pago Para Una Factura

- **Historia de usuario #41.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder registrar un pago para una factura específica con la finalidad de actualizar el estado de la deuda del estudiante.

- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir el ingreso de la fecha del pago, la selección del método de pago (Efectivo o Transferencia), el ingreso del monto abonado en dólares (USD), el ingreso del monto abonado en bolívares (VES) de ser necesario, el ingreso de la tasa de cambio al Banco Central de Venezuela de ser necesario y el ingreso de una referencia del pago o billete.
 - Registrar automáticamente la fecha de creación y de actualización.
 - Verificar que el pago registrado no sea mayor al monto de la factura y al monto de la deuda.
 - Actualizar, automáticamente, el monto de la deuda tras registrar el pago.
- **Requerimientos funcionales:**
 - Interfaz de usuario para registrar pagos.
 - Validación de datos de entrada.
 - Notificación de éxito o de error al momento de crear.
- **Limitaciones:** Si el estudiante no tiene deuda o no tiene productos agregados a la factura, no se podrá registrar un pago.
- **Dependencia:** #39.
- **Comentario:** Los comentarios no son obligatorios, al igual que la tasa en bolívares y del Banco Central de Venezuela no son campos obligatorios.

4.2.42 Generar Un Reporte De Calificaciones De Un Estudiante Por Semestre

- **Historia de usuario #42.**
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder generar un reporte de calificaciones de un estudiante específico por semestre para revisar el desempeño académico del estudiante durante un período determinado.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe mostrar todas las asignaturas cursadas por el estudiante en los semestres inscritos y el rango de duración del semestre en que la cursó, y la fecha de inicio y la fecha de finalización del semestre.
 - Debe mostrar las calificaciones obtenidas en número y en letra para cada asignatura.
 - Debe incluir la fecha de registro y el estatus de cada calificación.
 - Debe permitir la exportación del reporte en formato PDF.
- **Requerimientos funcionales:**
 - Botón para generar el reporte.
 - Visualización de calificaciones por asignatura y por semestre.
 - Exportación del reporte en PDF.
- **Limitaciones:** Ninguna.
- **Dependencia:** #7 y #30.
- **Comentario:** Si el estudiante no tiene asignaturas registradas, el botón debe ocultarse.

4.2.43 Generar Un Reporte De Pagos Efectuados En Un Mes Específico

- **Historia de usuario #43.**
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder generar un reporte de todos los pagos efectuados en un mes específico para revisar y para auditar los ingresos del periodo seleccionado.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe permitir la selección de un mes y de un año específicos.
 - Debe listar todos los pagos efectuados durante el mes seleccionado.
 - Debe incluir detalles de cada pago: fecha, monto en USD, monto en VES, método de pago y referencia.
 - Debe mostrar el total de ingresos por mes.
 - Debe permitir la exportación del reporte en formato PDF y Excel.
- **Requerimientos funcionales:**
 - Interfaz de usuario para generar el reporte.
 - Opción para seleccionar mes y año.
 - Visualización detallada de los pagos.
 - Exportación del reporte en PDF y en Excel para generar el reporte.
- **Limitaciones:** Ninguna.
- **Dependencia:** #41.
- **Comentario:** Si no hay pagos registrados en el mes y en el año seleccionado, debe mostrar un mensaje en pantalla notificando.

4.2.44 Generar Un Reporte De Estudiantes Matriculados Por Carrera

- **Historia de usuario #44.**
- **Usuario:** Administrador del sistema.

- **Descripción:** Como administrador del sistema, quiero poder generar un reporte de todos los estudiantes matriculados por carrera para obtener una visión general de la distribución de estudiantes en las diferentes carreras ofrecidas.
- **Prioridad:** Alta.
- **Criterios de aceptación:**
 - Debe listar todos los estudiantes matriculados por carrera.
 - Debe incluir detalles de cada estudiante: nombre, apellido, C.I./Pasaporte, fecha de matriculación y estado de la matrícula.
 - Debe permitir la exportación del reporte en formato PDF y Excel.
 - Debe registrar la fecha de generación del reporte.
- **Requerimientos funcionales:**
 - Interfaz de usuario para generar el reporte.
 - Visualización detallada de los estudiantes matriculados.
 - Exportación del reporte en PDF y en Excel.
- **Limitaciones:** Ninguna.
- **Dependencia:** #23.
- **Comentario:** Ninguno.

4.2.45 Generar Un Reporte De Todas Las Facturas

- **Historia de usuario #45.**
- **Usuario:** Administrador del sistema.
- **Descripción:** Como administrador del sistema, quiero poder generar un reporte de todas las facturas emitidas para revisar y para gestionar los cobros y los estados de pago.
- **Prioridad:** Alta.

- **Criterios de aceptación:**
 - Debe listar todas las facturas emitidas.
 - Debe incluir detalles de cada factura: número de factura, estudiante, fecha de emisión, estado, monto total y saldo.
 - Debe permitir la exportación del reporte en formato PDF y en Excel.
- **Requerimientos funcionales:**
 - Interfaz de usuario para generar el reporte.
 - Visualización detallada de las facturas.
 - Exportación del reporte en PDF y en Excel.
- **Limitaciones:** Ninguna.
- **Dependencia:** #39.
- **Comentario:** Ninguno.

www.bdigital.ula.ve

4.3 Product Backlog

Un Product Backlog es una lista priorizada de todas las tareas, las características y los requisitos que se necesitan para completar un proyecto, y es una herramienta fundamental en la metodología Scrum que está en constante evolución. Cada elemento del Product Backlog se conoce como historia de usuario.

Para este proyecto, se implementó el Product Backlog de las tablas #6, #7, #8 y #9, #10 y #11 en donde se contemplan todas las historias de usuario a desarrollar.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
1	Configuración del proyecto en Django	Configuración	*Creación del proyecto Django. *Configuración inicial del archivo settings.py. *Configuración del entorno virtual. *Instalación de dependencias. *Configuración de archivos urls.py y wsgi.py.	Alta	Ninguna
2	Crear aplicaciones modulares en el proyecto Django	Configuración	*Creación de las apps individuales. *Organización de modelos, de vistas, de urls y de templates dentro de cada app. *Registro de las apps en settings.py. *Inclusión de archivo admin.py en cada app.	Alta	#1
3	Establecer la arquitectura de ficheros del proyecto Django	Configuración	*Organización de directorios y de ficheros. *Inclusión de directorios templates con carpetas de cada app y un static general. *Configuración de archivo requirements.txt.	Alta	#1
4	Configurar el archivo settings.py del proyecto Django	Configuración	*Configuración de base de datos. *Configuración de zonas horarias y de idioma. *Configuración de archivos estáticos y de media. *Registro de aplicaciones instaladas. *Configuración de seguridad. *Configuración de middleware.	Alta	#1
5	Configuración inicial de la Base de Datos	Configuración	*Configuración de la conexión a la base de datos en el entorno de desarrollo. *Creación de modelos iniciales y de sus relaciones. *Migraciones aplicadas correctamente y tablas generadas en MySQL.	Alta	#2 y #3

Tabla 6#. Product Backlog Fuente: Propia.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
6	Sistema de Login	Login	*Página de Inicio de Sesión: Crear una vista y una plantilla sencilla para el formulario de inicio de sesión. *Autenticación: Implementar la lógica de autenticación utilizando el sistema de autenticación de Django (authenticate, login). *Manejo de Errores: Mostrar mensajes de error claros y específicos si el inicio de sesión falla. *Redirección: Redirigir al usuario a la página correspondiente después de iniciar sesión.	Alta	#5
7	Crear un nuevo estudiante	Estudiantes	*Formulario de Registro: Crear un formulario para la entrada de datos del estudiante. *Validación de Datos: Implementar validaciones para la identificación del estudiante y otros campos obligatorios. *Confirmación y Redirección: Mostrar mensaje de éxito y redirigir a la lista de estudiantes.	Alta	#2
8	Actualizar información del estudiante	Estudiantes	*Formulario de Edición: Crear una vista utilizando el mismo formulario para editar los datos del estudiante. *Validación de Datos: Implementar validaciones para asegurar que la identificación del estudiante sigue siendo única. *Registro de Fecha de Actualización: Actualizar automáticamente el campo update_day.	Media	#7
9	Eliminar estudiante	Estudiantes	*Confirmación de Eliminación: Mostrar un cuadro de diálogo para confirmar la eliminación. *Manejo de Relaciones: Asegurar que los datos relacionados se manejen correctamente (eliminar en cascada o mantener integridad referencial). *Confirmación de Éxito: Mostrar mensaje de éxito después de la eliminación.	Baja	#7
10	Visualizar estudiante	Estudiantes	*Vista Detallada: Crear una página de detalle para mostrar la información del estudiante. *Acceso a Datos Relacionados: Incluir enlaces o secciones para visualizar números de teléfono, correos electrónicos y nacionalidad.	Alta	#7

Tabla 7#. Product Backlog Fuente: Propia.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
11	Crear una carrera nueva	Plan de Formación	*Interfaz de usuario para ingresar los datos de la carrera. *Validaciones para los campos de nombre, de unidades de crédito y de duración.	Alta	#2
12	Actualizar una carrera	Plan de Formación	*Interfaz de usuario para modificar los datos de una carrera existente. *Validaciones para asegurar que los cambios sean coherentes.	Media	#11
13	Crear una cohorte	Plan de Formación	*Interfaz de usuario para ingresar los datos de la cohorte. *Validaciones para asegurar la unicidad del nombre dentro de la carrera.	Alta	#2
14	Calcular fecha posible de culminación	Plan de Formación	*Lógica de negocio para calcular la fecha posible de culminación. *Interfaz de usuario para mostrar y para editar la fecha de culminación junto con la cohorte.	Alta	#13
15	Definir un semestre actual	Plan de Formación	*Interfaz de usuario para activar un semestre.	Alta	#2
16	Actualizar fechas del semestre	Plan de Formación	*Interfaz de usuario para editar las fechas del semestre.	Alta	#15
17	Registrar una asignatura	Plan de Formación	*Interfaz de usuario para ingresar los datos de la materia. *Validaciones para asegurar la unicidad del código y el rango de las unidades de crédito.	Alta	#2
18	Asociar una asignatura a una carrera	Plan de Formación	*Interfaz de usuario para asociar asignaturas a planes de estudio. *Validaciones para asegurar que la materia pertenezca a la carrera correcta.	Alta	#11 y #17
19	Crear un plan de estudio	Plan de Formación	*Interfaz de usuario para asociar asignaturas a planes de estudio. *Validaciones para asegurar que la materia pertenezca a la carrera correcta.	Alta	#11
20	Actualizar un plan de estudio	Plan de Formación	*Interfaz de usuario para modificar los datos de un plan de estudios existente. *Validaciones para asegurar que los cambios sean coherentes.	Media	#19

Tabla 8#. Product Backlog Fuente: Propia.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
21	Asignar una asignatura a un semestre y a un plan de estudio	Plan de Formación	*Interfaz de usuario para asignar materias a semestres. *Validaciones para asegurar la correcta asociación de materias y de semestres.	Alta	#17 y #19
22	Actualizar asignatura asociada a un plan de estudio	Plan de Formación	*Interfaz de usuario para actualizar la etapa del semestre de la asignatura respecto al plan de estudio.	Media	#21
23	Matricular un estudiante	Plan de Formación	*Interfaz de usuario para matricular estudiantes. *Validaciones para asegurar la correcta asociación del estudiante al plan de estudios y a la cohorte.	Alta	#7, #13 y #19
24	Actualizar el estatus del estudiante en su matriculación	Plan de Formación	*Interfaz de usuario para actualizar el estatus del estudiante en la matriculación de un plan de estudio. *Validaciones para asegurar que el cambio de estatus sea coherente.	Alta	#23
25	Registrar la inscripción de un estudiante en una materia	Registro	*Interfaz de usuario para inscribir a un estudiante en una asignatura. *Validaciones para asegurar la correcta asociación y la unicidad de la inscripción.	Alta	#7, #15 y #18
26	Actualizar la inscripción de un estudiante	Registro	*Interfaz de usuario para modificar la inscripción de un estudiante. *Validaciones para asegurar que los cambios sean coherentes y se mantenga la unicidad.	Media	#25
27	Visualizar todas las inscripciones de un estudiante	Estudiantes	*Interfaz de usuario para visualizar todas las inscripciones de un estudiante.	Alta	#7 y #25
28	Registrar calificación de una asignatura	Registro	*Interfaz de usuario para registrar calificaciones. *Validaciones para asegurar la correcta asociación de calificaciones con inscripciones.	Alta	#25
29	Actualizar una calificación existente	Registro	*Interfaz de usuario para modificar calificaciones y estados de materias. *Validaciones para asegurar que los cambios sean coherentes.	Media	#28
30	Visualizar el historial de calificaciones de un estudiante	Estudiantes	*Interfaz de usuario para visualizar el historial de calificaciones de un estudiante.	Alta	#7 y #28

Tabla 9#. Product Backlog Fuente: Propia.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
31	Registrar un nuevo arancel, curso o unidad curricular	Productos	*Interfaces de usuario independientes para registrar nuevos aranceles, cursos o unidades curriculares. *Validaciones para asegurar la unicidad del nombre. *Notificación de éxito o de error al momento de crear.	Alta	#2
32	Actualizar un nuevo arancel, curso o unidad curricular	Productos	*Interfaz de usuario para modificar la descripción de un arancel, de un curso o de una unidad curricular. *Notificación de éxito o de error al momento de actualizar.	Baja	#31
33	Registrar el precio de un arancel para una nacionalidad específica	Productos	*Interfaz de usuario para registrar precios de aranceles. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de crear.	Alta	#31
34	Actualizar el precio de un arancel para una nacionalidad específica	Productos	*Interfaz de usuario para actualizar precios de aranceles. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de actualizar.	Alta	#33
35	Registrar el precio de un curso para una nacionalidad específica	Productos	*Interfaz de usuario para registrar precios de aranceles. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de crear.	Alta	#31
36	Actualizar el precio de un curso para una nacionalidad específica	Productos	*Interfaz de usuario para actualizar precios de cursos. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de actualizar.	Alta	#35
37	Registrar el precio de una unidad curricular para una nacionalidad específica	Productos	*Interfaz de usuario para registrar precios de unidades curriculares. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de crear.	Alta	#31
38	Actualizar el precio de una unidad curricular para una nacionalidad específica	Productos	*Interfaz de usuario para registrar precios de unidades curriculares. *Validaciones para asegurar la correcta asociación. *Notificación de éxito o de error al momento de crear.	Alta	#37
39	Crear una nueva factura para un estudiante	Facturas	*Interfaz de usuario para crear nuevas facturas. *Generación automática de números de factura únicos. *Validación de datos de entrada. *Notificación de éxito o de error al momento de crear.	Alta	#2 y #7
40	Agregar un elemento a una factura existente	Facturas	*Interfaz de usuario para agregar elementos a una factura. *Validación de datos de entrada. *Notificación de éxito o de error al momento de crear.	Alta	#33, #35, #37 y #39

Tabla 10#. Product Backlog Fuente: Propia.

Product Backlog					
Nº	Historia de Usuario	Modulo	Requerimiento Funcional	Prioridad	Dependencia
41	Registrar un pago para una factura	Facturas	*Interfaz de usuario para registrar pagos. *Validación de datos de entrada. *Notificación de éxito o de error al momento de crear.	Alta	#39
42	Generar un reporte de calificaciones de un estudiante por semestre	Reportes	*Botón para generar el reporte. *Visualización de calificaciones por asignatura y por semestre. *Exportación del reporte en PDF.	Alta	#7 y #30
43	Generar un reporte de pagos efectuados en un mes específico	Reportes	*Interfaz de usuario para generar el reporte. *Opción para seleccionar mes y año. *Visualización detallada de los pagos. *Exportación del reporte en PDF y en Excel para generar el reporte.	Alta	#41
44	Generar un reporte de estudiantes matriculados por carrera	Reportes	*Interfaz de usuario para generar el reporte. *Visualización detallada de los estudiantes matriculados. *Exportación del reporte en PDF y en Excel.	Alta	#23
45	Generar un reporte de todas las facturas	Reportes	*Interfaz de usuario para generar el reporte. *Visualización detallada de las facturas. *Exportación del reporte en PDF y en Excel.	Alta	#39

Tabla 11#. Product Backlog Fuente: Propia.

4.4 Sprint Backlog

El Sprint Backlog es un componente clave en la metodología Scrum utilizado ampliamente en el desarrollo ágil de software. Este es una lista detallada de tareas y de elementos del Product Backlog en donde hay un compromiso por parte de los desarrolladores para completarlas durante un Sprint (un período de trabajo definido; generalmente, de 1 a 4 semanas).

Este proyecto se desarrolló siguiendo los Sprints que se incluyen en las tablas #12 y #13.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
1	2 semanas	#1: Configuración del proyecto en Django	Configuración	1 día
		#2: Crear aplicaciones modulares en el proyecto Django	Configuración	3 días
		#3: Establecer la arquitectura de ficheros del proyecto Django	Configuración	1 semana
		#4: Configurar el archivo settings.py del proyecto Django	Configuración	1 día
		#5: Configuración inicial de la Base de Datos	Configuración	1 semana
2	1 semana	#6: Sistema de Login	Login	2 días
		#7: Crear un nuevo estudiante	Estudiantes	1 día
		#8: Actualizar información del estudiante	Estudiantes	0,5 días
		#9: Eliminar estudiante	Estudiantes	0,5 días
		#10: Visualizar estudiante	Estudiantes	1 día
3	3 semanas	#11: Crear una carrera nueva	Plan de Formación	1 día
		#12: Actualizar una carrera	Plan de Formación	0,5 días
		#13: Crear una cohorte	Plan de Formación	1 día
		#14: Calcular fecha posible de culminación	Plan de Formación	1 día
		#15: Definir un semestre actual	Plan de Formación	0,5 días
		#16: Actualizar fechas del semestre	Plan de Formación	0,5 días
		#17: Registrar una asignatura	Plan de Formación	0,5 días
		#18: Asociar una asignatura a una carrera	Plan de Formación	1 día
		#19: Crear un plan de estudio	Plan de Formación	1 día
		#20: Actualizar un plan de estudio	Plan de Formación	0,5 días
		#21: Asignar una asignatura a un semestre y a un plan de estudio	Plan de Formación	0,5 días
		#22: Actualizar asignatura asociada a un plan de estudio	Plan de Formación	0,5 días
		#23: Matricular un estudiante	Plan de Formación	1 semana
		#24: Actualizar el estatus del estudiante en su matriculación	Plan de Formación	1,5 días

Tabla #12. Sprint Backlog Fuente: Propia.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
4	2 semanas	#25: Registrar la inscripción de un estudiante en una materia	Registro	3 días
		#26: Actualizar la inscripción de un estudiante	Registro	2 días
		#27: Visualizar todas las inscripciones de un estudiante	Estudiantes	1 día
		#28: Registrar calificación de una asignatura	Registro	3 días
		#29: Actualizar una calificación existente	Registro	2 días
		#30: Visualizar el historial de calificaciones de un estudiante	Estudiantes	1 día
5	2 semanas	#31: Registrar un nuevo arancel / curso / unidad curricular	Productos	0,5 días
		#32: Actualizar un nuevo arancel / curso / unidad curricular	Productos	0,5 días
		#33: Registrar el precio de un arancel para una nacionalidad específica	Productos	0,5 días
		#34: Actualizar el precio de un arancel para una nacionalidad específica	Productos	0,5 días
		#35: Registrar el precio de un curso para una nacionalidad específica	Productos	0,5 días
		#36: Actualizar el precio de un curso para una nacionalidad específica	Productos	0,5 días
		#37: Registrar el precio de una unidad curricular para una nacionalidad específica	Productos	1 semana
		#38: Actualizar el precio de una unidad curricular para una nacionalidad específica	Productos	3 días
6	2 semanas	#39: Crear una nueva factura para un estudiante	Facturas	2 días
		#40: Agregar un elemento a una factura existente	Facturas	1 semana
		#41: Registrar un pago para una factura	Facturas	1 semana
7	1 semana	#42: Generar un reporte de calificaciones de un estudiante por semestre	Reportes	2 días
		#43: Generar un reporte de pagos efectuados en un mes específico	Reportes	1 día
		#44: Generar un reporte de estudiantes matriculados por carrera	Reportes	1 día
		#45: Generar un reporte de todas las facturas	Reportes	1 día

Tabla #13. Sprint Backlog Fuente: Propia.

Capítulo V

Propuesta De Solución

En este capítulo, se aborda la propuesta de solución a los requerimientos expuestos en el capítulo anterior, de modo que se explica la arquitectura elegida para el desarrollo de la aplicación web, así como las herramientas de *software* utilizadas para la ejecución.

5.1 Arquitectura A Utilizar

Para el desarrollo de la aplicación web, se utilizará la arquitectura MVT (Model-View-Template) proporcionada por Django.

La arquitectura MVT (Modelo-Vista-Template) es un patrón de diseño utilizado en el *framework* Django para el desarrollo de aplicaciones web. Las fases en un navegador web se desarrollan de la siguiente manera: primero, el navegador envía una solicitud HTTP al servidor. Django procesa esta solicitud utilizando la “Vista” correspondiente, que interactúa con el “Modelo” para obtener los datos necesarios. Estos datos se pasan a un “Template”, que se encarga de renderizar la información en un formato HTML. Finalmente, el servidor envía la respuesta HTML de vuelta al navegador, donde se muestra al usuario. Este ciclo asegura una separación clara entre la lógica del servidor y la presentación de la interfaz de usuario.

A continuación, se definen los conceptos de vista, de modelo y de *template*:

Vista (View): MDN Web Docs¹⁵ (s.f.) señala que:

¹⁵Mozilla Developer Network. (n.d.). Introducción a Django. MDN Web Docs. Recuperado el 14 de mayo de 2024, de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos y delegan el formateo de la respuesta a las plantillas.

Modelos (Models): de acuerdo a (MDN Web Docs, s.f.), los modelos son objetos de Python que “definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y para consultar registros en la base de datos”.

Plantilla (Templates): (MDN Web Docs, s.f.) explica que una plantilla “es un fichero que define la estructura o diagrama de otro fichero (tal como una página de HTML) con marcadores de posición que se utilizan para representar el contenido real”.

URL’s: Como explica MDN Web Docs (s.f.):

Aunque es posible procesar peticiones de cada URL individual vía una función individual, es mucho más sostenible escribir una función de visualización separada para cada recurso. Se usa un mapeador URL para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL se usa para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL puede, también, emparejar patrones de cadenas o dígitos específicos que aparecen en una URL y los pasan a la función de visualización como datos.

5.2 Herramientas A Utilizar

Para la implementación del sistema se hizo uso de herramientas de desarrollo web. Toda la base de la aplicación se hizo utilizando el framework Django, el cual es un marco de trabajo desarrollado en Python y una herramienta que se puede usar para el desarrollo *full-stack* de aplicaciones y de páginas web. También, se utilizó el gestor de base de datos MySQL, así como la herramienta MySQL Workbench. Para el entorno de desarrollo, se utilizó Visual Studio Code

y, para la carga dinámica de datos, se implementó AJAX. Como marco de diseño se utilizó Bootstrap 5.

A continuación, se explicará cada herramienta utilizada para el desarrollo del producto final con detalle.

5.2.1 Django

Amazon Web Services (s.f.)¹⁶ define Django como un *software* que “puede utilizarse para desarrollar aplicaciones web de forma rápida y eficiente. La mayoría de aplicaciones web tienen varias funciones comunes, como la autenticación, la recuperación de información de una base de datos y la administración de *cookies*”.

Amazon Web Services (s.f.) señala que Django administra el código de solicitud y de respuesta del servidor y del cliente utilizando una arquitectura Modelo-Vista-Template que se explicará a continuación.

Modelo: De acuerdo a Amazon Web Services (s.f.):

Los modelos de Django actúan como interfaz entre la base de datos y el código del servidor. Son la única fuente definitiva de información sobre sus datos. Estos modelos de datos contienen los campos y las operaciones esenciales que necesita para interactuar con la base de datos.

Vista: Amazon Web Services (s.f.) explica que:

Las vistas de Django procesan la solicitud utilizando los modelos. Se puede escribir una función para cada tipo de solicitud que los visitantes de un sitio pueden hacer a este. Una función de vista puede tomar la solicitud como entrada y devolver una respuesta, que puede ser un código de error, una imagen, un archivo o cualquier tipo de dato.

¹⁶ Amazon Web Services, Inc. (n.d.). ¿Qué es Django? AWS. Recuperado el 14 de mayo de 2024, de <https://aws.amazon.com/es/what-is/django/>

Template: “las plantillas de Django administran la presentación de la página web en el navegador”. (Amazon Web Services, s.f.)

5.2.3 Aplicación de Django

Microsoft (2023)¹⁷ define una aplicación de Django como:

Un paquete de Python independiente que contiene un conjunto de archivos relacionados para propósito específico. Un proyecto de Django puede contener muchas aplicaciones, lo que ayuda a un *host web* a atender muchos puntos de entrada independientes desde un solo nombre de dominio.

5.2.4 Visual Studio Code

Visual Studio Code¹⁸ es un editor de código fuente ligero, pero eficaz, que se ejecuta en el escritorio y está disponible para Windows, para macOS y para Linux. Incluye compatibilidad integrada con JavaScript, con Type Script y con Node.js., y cuenta con un amplio ecosistema de extensiones para otros lenguajes y entornos de ejecución (como C++, C#, Java, Python, Go, .NET).

5.2.5 MySQL Workbench

MySQL Workbench¹⁹ es una herramienta visual unificada para arquitectos de bases de datos, para desarrolladores y para administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, para la administración de usuarios, para la copia de seguridad y para mucho más.

¹⁷ Microsoft. (n.d.). Paso 2: Crear una aplicación en Django en Visual Studio. Microsoft Learn. Recuperado el 21 de mayo de 2024, de <https://learn.microsoft.com/es-es/visualstudio/python/...>

¹⁸ Microsoft. (s.f.). Visual Studio. Recuperado de <https://visualstudio.microsoft.com/es/>

¹⁹ Oracle Corporation. (s.f.). MySQL Workbench. Recuperado de <https://www.mysql.com/products/workbench/>

5.2.6 AJAX

AJAX ²⁰(Asynchronous JavaScript and XML) “es una práctica de programación utilizada para construir páginas web más complejas y dinámicas utilizando una tecnología conocida como XMLHttpRequest”. (MDN Web Docs, s.f.).

Tal y como explica MDN Web Docs (s.f.), AJAX permite:

Modificar partes específicas del DOM de una página HTML sin la necesidad de refrescar la página entera. AJAX, también, permite trabajar asíncronamente; esto significa que el código seguirá corriendo mientras esa parte de la página del sitio web intenta recargarse.

www.bdigital.ula.ve

²⁰ Mozilla. (s.f.). AJAX. Recuperado de <https://developer.mozilla.org/es/docs/Glossary/AJAX>

CAPÍTULO VI

Análisis De Resultados

En este capítulo, se detalla el desarrollo del proyecto a través de una serie de sprints enfocados en un aspecto clave de la implementación. Inicialmente, en el Sprint 1, se realizó la configuración inicial del proyecto, asegurando la preparación del entorno de desarrollo y la estructuración del proyecto con Django. El Sprint 2 se dedicó a la autenticación de usuarios previamente registrados y se trabajó en el módulo de estudiantes. Continuando con el desarrollo, el Sprint 3 consistió en todo lo relacionado con el módulo de plan de formación. En el Sprint 4, se abordó el módulo de inscripción de asignaturas, el registro y la consulta de calificaciones, facilitando el seguimiento académico de cada estudiante matriculado. El Sprint 5 se enfocó en el módulo de productos, desde aranceles, cursos y unidades curriculares, implementando la funcionalidad para administrarlos eficientemente. Posteriormente, el Sprint 6 estuvo orientado a la creación del módulo de facturas para facturar productos a los estudiantes tabulados a su nacionalidad previamente asignada. Finalmente, en el Sprint 7, se ejecutó la generación de reportes académicos y administrativos, esenciales para la toma de decisiones. Estos sprints permitieron estructurar y desarrollar el proyecto de manera organizada y eficiente, asegurando el cumplimiento de los objetivos planteados.

6.1 Sprint N.º 1

En el Sprint 1, se llevó a cabo la configuración inicial del proyecto, la cual es fundamental para asegurar que todos los componentes del sistema funcionen de manera correcta y estén distribuidos los archivos de manera ordenada. Al completar estas historias, se tuvo ya una estructura inicial para ir creando las distintas funcionalidades del proyecto mostradas en la tabla

#14. Además, se creó, inicialmente, la base de datos dentro del aplicativo Workbench para migrar todos los modelos creados en las distintas aplicaciones del proyecto en Django.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
1	2 semanas	#1: Configuración del proyecto en Django	Configuración	1 día
		#2: Crear aplicaciones modulares en el proyecto Django	Configuración	3 días
		#3: Establecer la arquitectura de ficheros del proyecto Django	Configuración	1 semana
		#4: Configurar el archivo settings.py del proyecto Django	Configuración	1 día
		#5: Configuración inicial de la Base de Datos	Configuración	1 semana

Tabla #14. Historias de usuario incorporadas en el Sprint N.º 1. Fuente: Propia.

6.1.1 Historia De Usuario #3: Establecer la arquitectura de ficheros del proyecto Django

En la Figura #3, podemos observar la estructura de ficheros y de las aplicaciones del proyecto. Este sprint incluye la creación de un entorno de desarrollo, la configuración de las bases de datos y la estructuración del proyecto utilizando el *framework* Django. Además, se crearon las aplicaciones básicas que serán la columna vertebral del proyecto.

```

app/
+---config
# +---__pycache__
+---core
# +---login
# +---migrations
# +---__pycache__
# +---__pycache__
+---private
# +---bills
# a +---migrations
# a +---__pycache__
# a +---__pycache__
# +---careers
# a +---migrations
# a +---__pycache__
# a +---__pycache__
+---products
# a +---migrations
# a +---__pycache__
# a +---__pycache__
# +---record
# a +---migrations
# a +---__pycache__
# a +---__pycache__
# +---reports
# a +---migrations
# a +---__pycache__
# a +---__pycache__
# +---student
# +---migrations
# a +---__pycache__
# a +---__pycache__
+---requirements

+---templates
+---bills
# +---invoice
# +---invoice_element
# +---payment
+---careers
# +---career
# +---cohort
# +---current_semester
# +---enroll_student
# +---matter
# +---semester_matter
# +---study_plan
+---layout
# +---partials
+---login
+---products
# +---courses
# +---curricular_units
# +---price_courses
# +---price_curricularunits
# +---price_tariff
# +---tariff
+---record
# +---grade_record
# +---student_registration
+---reports
+---student

```

Figura #3. Estructura de ficheros del proyecto Django. Fuente: Propia.

6.1.2 Historia De Usuario #5: Configuración inicial de la Base de Datos

El diagrama Entidad-Relación (ER) presentado en la Figura #4, describe la estructura de una base de datos diseñada para la gestión académica y administrativa de la aplicación web. Este modelo abarca múltiples aspectos del sistema, incluyendo la inscripción de estudiantes, la gestión de carreras, los planes de estudio, las nacionalidades de estudiantes, los números de teléfono y los correos electrónicos de estudiantes, así como registros de calificaciones y productos para facturar a los estudiantes.

Las principales entidades en el diagrama incluyen `student`, `careers`, `products` y `bills`, cada una con sus respectivas subentidades y relaciones. Por ejemplo, la entidad `student` se relaciona con `student\_phone`, `student\_email`, y `student\_studentnationality`, reflejando, así, los datos personales y administrativos del estudiante. Las carreras y materias se gestionan a través de entidades como `careers\_career`, `careers\_matter`, y `careers\_enrollstudent`, que organizan la inscripción y el seguimiento académico.

Además, la aplicación web incluye módulos para la facturación y la gestión de pagos, representados por entidades como `bills\_invoice`, `bills\_invoiceelement`, y `bills\_payment`. Estas entidades están vinculadas a los productos académicos como `products\_courses` y `products\_curricularunits`, permitiendo una gestión integral de los costos y de los servicios ofrecidos por la institución.

Para las historias de usuario #1, #2 y #4 se cumplieron con los requisitos funcionales y se lograron los criterios de aceptación. En la Figura #3, se muestra la carpeta config, en dónde se encuentra la configuración del archivo setting. Además, se muestran las distintas apps creadas.

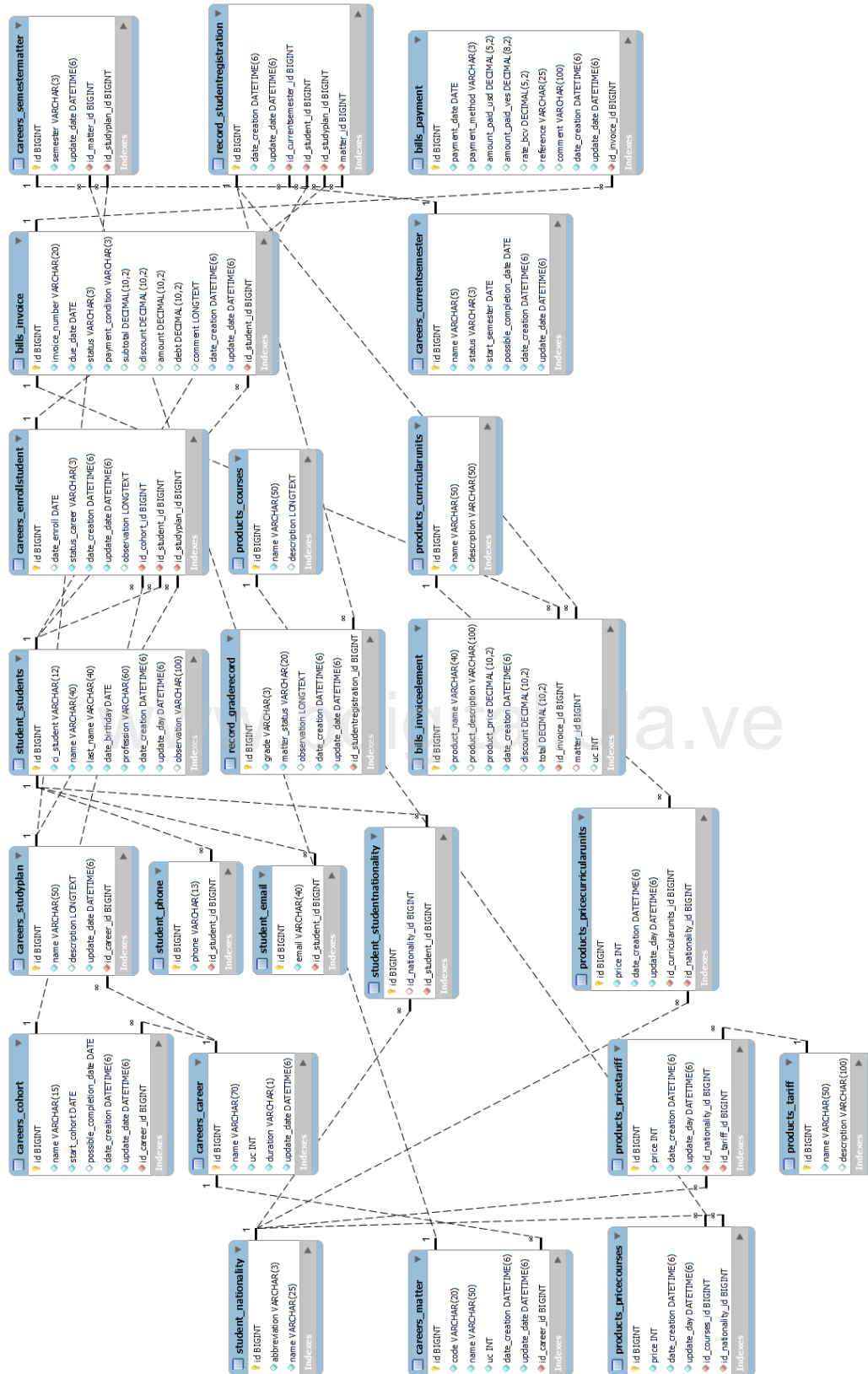


Figura #4. Diagrama Entidad-Relación del modelo de la base de datos. Fuente: Propia.

6.2 Sprint N.º 2

Este sprint se centró en la creación y en el desarrollo del módulo de inicio de sesión, o Login/Logout, junto al desarrollo del módulo de estudiantes. En la tabla #15, se observan las historias de usuario desarrolladas en esta etapa.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
2	1 semana	#6: Sistema de Login	Login	2 días
		#7: Crear un nuevo estudiante	Estudiantes	1 día
		#8: Actualizar información del estudiante	Estudiantes	0,5 días
		#9: Eliminar estudiante	Estudiantes	0,5 días
		#10: Visualizar estudiante	Estudiantes	1 día

Tabla #15. Historias de usuario incorporadas en el Sprint N.º 2. Fuente: Propia.

6.2.1 Historia De Usuario #6: Sistema De Login

La aplicación web permite el ingreso mediante la autenticación y la autorización. En la Figura #5, se observa el formulario de inicio de sesión, en el que se solicita un nombre de usuario y la contraseña previamente generada. Se incluye la notificación para informar al usuario que uno de los datos de acceso ingresados es inválido, lo que se logra apreciar en la Figura #6.



Figura #5. Inicio de sesión en la aplicación web. Fuente: Propia.



Figura #6. Notificación cuando se ingresa un dato incorrecto en el inicio de sesión. Fuente: Propia.

6.2.2 Historia De Usuario #7: Crear Un Nuevo Estudiante

Se diseñó e implementó el formulario para crear un nuevo estudiante con los datos requeridos y, en la visualización, se añaden botones para actualizar y para eliminar al estudiante seleccionado. En la figura #6, se pueden observar los campos que se deben llenar para crear un estudiante en la aplicación web.

Formulario para crear un nuevo estudiante

Datos del Estudiante

Nacionalidad*
 Seleccione una nacionalidad

Número de Identificación*
 CI o Pasaporte

Nombres*
 Nombres del estudiante

Apellidos*
 Apellidos del estudiante

Fecha de nacimiento*
 dd/mm/aaaa

Profesión*
 Profesion o especialidad del estudiante.

Observacion
 Redacte alguna observación relevante de ser necesario.

Datos de contacto

Numero de telefono*
 Numero de telefono

Correo electronico*
 Correo electronico

Numero de telefono*
 Numero de telefono

Correo electronico*
 Correo electronico

Crear estudiante **Cancelar**

Figura

#7. Formulario para crear un nuevo estudiante. Fuente: Propia.

6.2.3 Historia De Usuario #8: Actualizar Información Del Estudiante

Luego de que el usuario registra la información de un nuevo estudiante, verifica si ya existe en la base de datos en el campo de C.I./Pasaporte, ya que es un dato único y no se puede repetir. De existir, se muestra una alerta o notificación de error al usuario, tal como muestra la Figura #8. En caso de estar todos los campos obligatorios llenos y de no existir dicho campo, se crea el estudiante y se redirige a la sección de búsqueda de estudiantes. Para la historia de usuario #9 creo la funcionalidad pero por reglas de negocio no se habilitó esta historia.

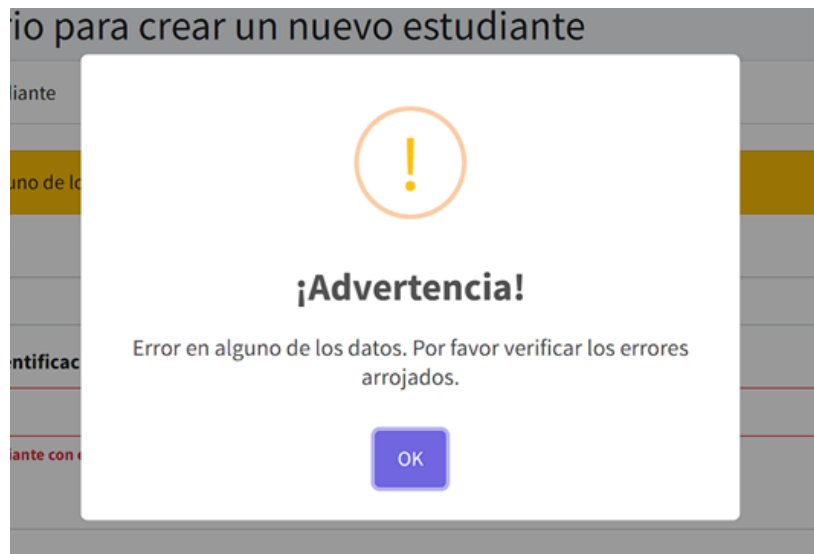


Figura #8 Notificación de error tras intentar crear un usuario con CI/Pasaporte ya existente. Fuente: Propia.

6.2.4: Historia De Usuario #10: Visualizar Estudiante

Para visualizar la información del estudiante, se utiliza un formulario de búsqueda que se muestra en la Figura #9. Luego de encontrar al estudiante, se ingresa a través del botón de visualización ubicado en la columna Acciones. En la Figura #10, se muestra que información se imprime del estudiante registrado al usuario.

Buscar Estudiante

Buscar un Estudiante

Buscar un estudiante por CI o Pasaporte

Buscar estudiante

Lista de estudiantes recién creados

Crear Estudiante +

CI o Pasaporte	Nombre	Apellido	Acciones
9471742	Albio	Marquez	
25152612	Gabriel Andres	Belandria Lopez	

Figura #9. Buscador de estudiantes por número de CI o Pasaporte. Fuente: Propia.

Información del estudiante

Datos del Estudiante Datos Académicos Facturas Estudiante: 25151141 Abraham Mousalli

← Atrás Actualizar estudiante

Datos Personales

Cedula/Pasaporte	Nombre	Apellido	Fecha de Nacimiento	Edad	Nacionalidad	Profesion	Observacion
25151141	Abraham	Mousalli	27 de mayo de 1995	29 años	Extranjero Resto del Mund	Estudiante.	

Datos de Contacto

Numero	Correo
04122055716	a@a.com

Datos de Actualización

Fecha de creación	Última actualización
27 de mayo de 2024 a las 21:09	29 de mayo de 2024 a las 10:00

Figura #10. Visualizar los datos almacenados del estudiante. Fuente: Propia.

6.3 Sprint N.º 3

Durante el Sprint 3, se desarrolló la experiencia del módulo de plan de formación. En este módulo, se incluye todo lo relacionado con los planes de estudios de posgrado que ofrece la SEPEC a sus estudiantes. En la Tabla #16, se encuentra el desglose de historias de usuario trabajadas en este sprint.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
3	3 semanas	#11: Crear una carrera nueva	Plan de Formación	1 día
		#12: Actualizar una carrera	Plan de Formación	0,5 días
		#13: Crear una cohorte	Plan de Formación	1 día
		#14: Calcular fecha posible de culminación	Plan de Formación	1 día
		#15: Definir un semestre actual	Plan de Formación	0,5 días
		#16: Actualizar fechas del semestre	Plan de Formación	0,5 días
		#17: Registrar una asignatura	Plan de Formación	0,5 días
		#18: Asociar una asignatura a una carrera	Plan de Formación	1 día
		#19: Crear un plan de estudio	Plan de Formación	1 día
		#20: Actualizar un plan de estudio	Plan de Formación	0,5 días
		#21: Asignar una asignatura a un semestre y a un plan de estudio	Plan de Formación	0,5 días
		#22: Actualizar asignatura asociada a un plan de estudio	Plan de Formación	0,5 días
		#23: Matricular un estudiante	Plan de Formación	1 semana
		#24: Actualizar el estatus del estudiante en su matriculación	Plan de Formación	1,5 días

Tabla #16. Historias de usuario incorporadas en el Sprint N.º 3. Fuente: Propia.

6.3.1: Historia De Usuario #11: Crear Una Carrera Nueva

Se comenzó la experiencia dentro de este módulo, creando una carrera, conocida como programa en la aplicación web también. En la Figura #11, se visualiza el formulario para crear una carrera.

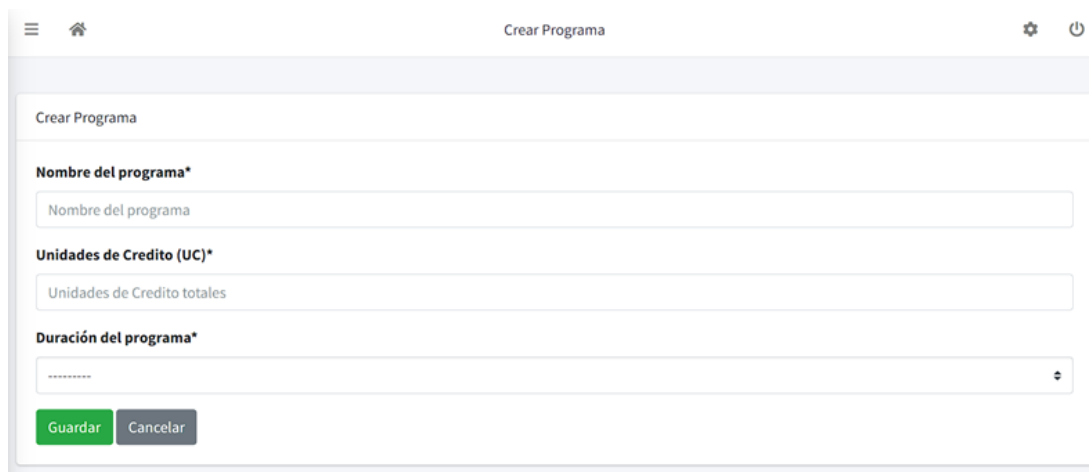


Figura #11. Formulario para crear un programa o una carrera. Fuente: Propia.

6.3.2: Historia De Usuario #12: Actualizar Una Carrera

En la Figura #12, el formulario para actualizar una carrera o programa. Tras seleccionar la duración del programa, se contempló este dato al momento de crear una cohorte y asociarla a un programa con la finalidad de calcular, automáticamente, la fecha de duración del programa para la cohorte seleccionada.

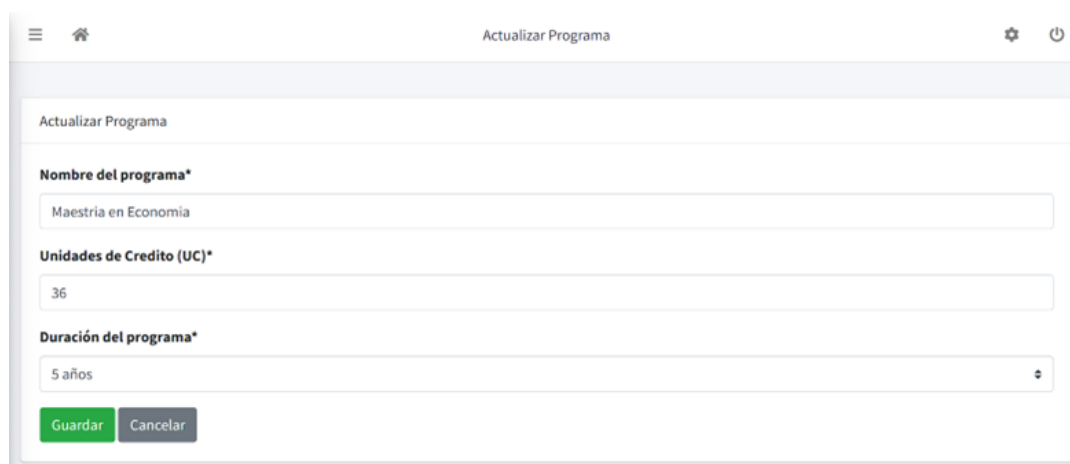
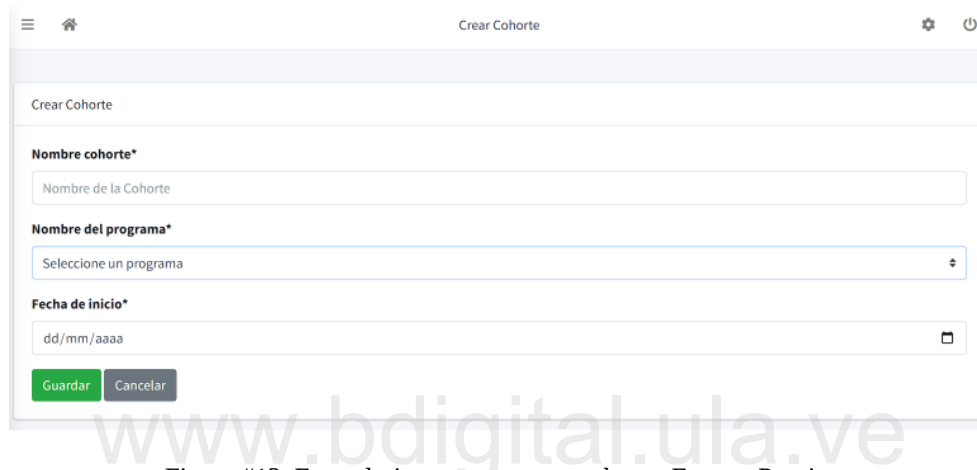


Figura #12. Formulario para actualizar un programa o una carrera. Fuente: Propia.

6.3.3: Historia De Usuario #13: Crear Una Cohorte

Para crear una cohorte, se requiere, previamente, haber creado un programa; así, se puede listar y seleccionar el programa con el cual se relaciona, y que le pertenece a la cohorte. En la Figura #13, se muestra el formulario para crear la cohorte y la figura #14 refleja el formulario que se debe llenar para actualizar una cohorte.

El formulario 'Crear Cohorte' tiene un encabezado con un menú de hamburguesa, un ícono de casa y el título 'Crear Cohorte'. A la derecha hay íconos de configuración y apagado. El cuerpo del formulario contiene tres campos obligatorios: 'Nombre cohorte*' con un input de texto, 'Nombre del programa*' con un menú desplegable que muestra 'Seleccione un programa', y 'Fecha de inicio*' con un calendario que muestra 'dd/mm/aaaa'. Al final hay dos botones: 'Guardar' (verde) y 'Cancelar' (gris).

Crear Cohorte

Nombre cohorte*

Nombre del programa*

Fecha de inicio*

Guardar Cancelar

Figura #13. Formulario para crear una cohorte. Fuente: Propia.

El formulario 'Actualizar Cohorte' tiene un encabezado con el título 'Actualizar Cohorte'. El cuerpo contiene cuatro campos: 'Nombre cohorte*' con el valor '2024', 'Nombre del programa*' con el valor 'Doctorado en Economía', 'Fecha de inicio*' con el valor '27/05/2024', y 'Fecha posible de culminacion' con el valor '26/05/2030'. Los campos de fecha tienen íconos de calendario. Al final hay dos botones: 'Guardar' (verde) y 'Cancelar' (gris).

Actualizar Cohorte

Nombre cohorte*

Nombre del programa*

Fecha de inicio*

Fecha posible de culminacion

Guardar Cancelar

Figura #14. Formulario para actualizar una cohorte. Fuente: Propia.

6.3.4: Historia De Usuario #14: Calcular Fecha Posible De Culminación

Se sobrescribió el método save en el modelo Cohort para calcular y para asignar una fecha de posible finalización (possible_completion_date) basada en la duración de la carrera

(id_career.duration) cuando se crea un nuevo objeto. Si el objeto ya existe, es decir, tiene una clave primaria pk, no recalcula o modifica la fecha de posible finalización.

```
def save(self, *args, **kwargs):
    if not self.pk:
        if self.id_career.duration:
            duration_years = int(self.id_career.duration)
            duration_days = duration_years * 365
            self.possible_completion_date = self.start_cohort + timedelta(days=duration_days)
    super().save(*args, **kwargs)
```

Figura #15. Método save sobrescrito para calcular la posible fecha de finalización. Fuente: Propia.

6.3.5: Historia De Usuario #15: Definir Un Semestre Actual

Al momento de crear un semestre, se necesita seleccionar el rango de fechas que se encontrará activo y cuando finaliza o se culmina ese semestre. En la Figura #16, se aprecia el formulario.

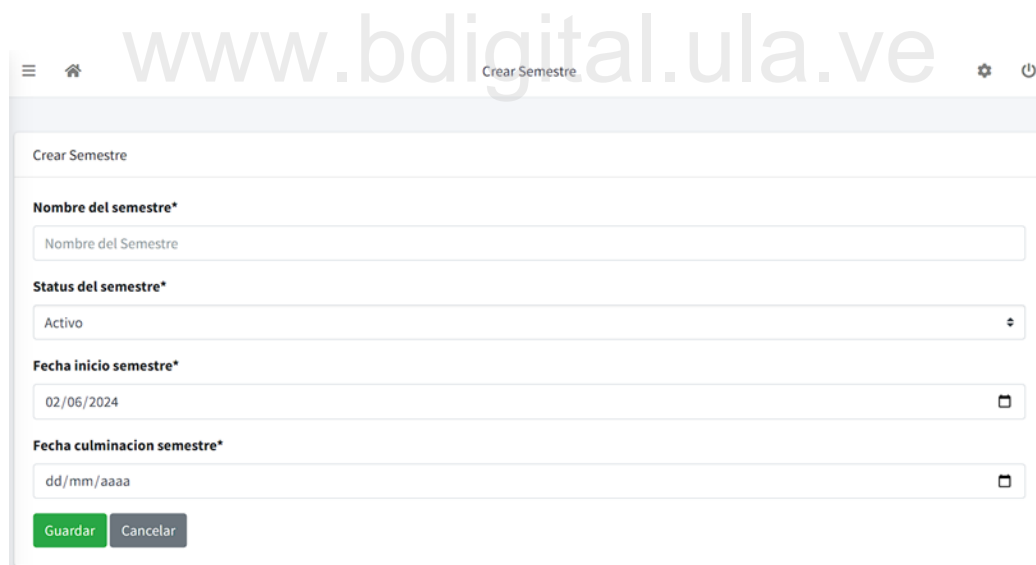


Figura #16. Formulario para crear un semestre. Fuente: Propia.

6.3.6: Historia De Usuario #16: Actualizar Fechas Del Semestre

Para actualizar, se necesita poder actualizar este rango de fechas, ya que las fechas son variables. En la Figura #17, se muestra la actualización de un semestre.



Figura #17. Formulario para actualizar un semestre. Fuente: Propia.

6.3.7: Historia De Usuario #17: Registrar Una Asignatura

Las asignaturas son una parte fundamental en la aplicación web, ya que estas asignaturas serán inscritas para cada estudiante, facturadas por la cantidad de unidades crédito (UC) que corresponda, visualizadas en EL reporte de calificaciones del estudiante. En la Figura #18, se visualiza el formulario para crear una asignatura.

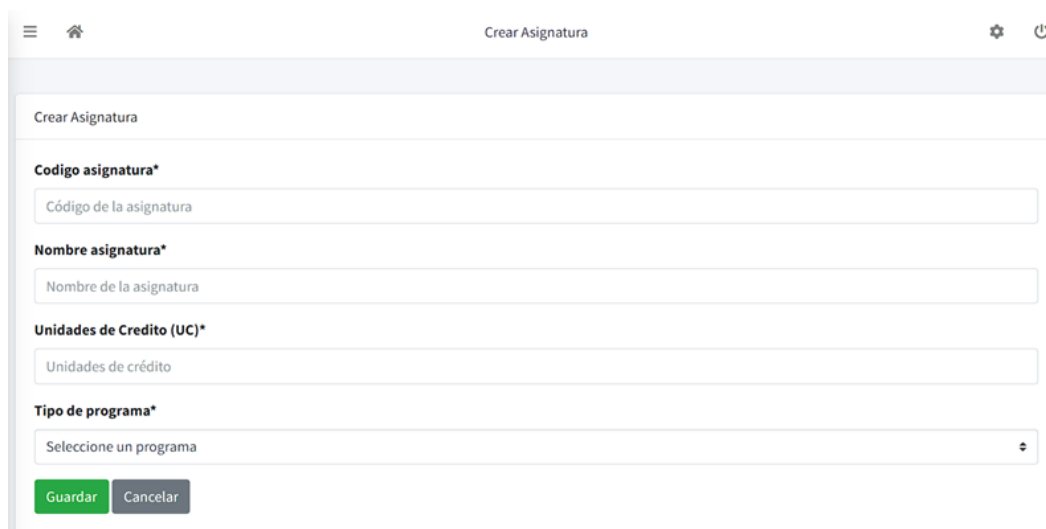


Figura #18. Formulario para crear una asignatura. Fuente: Propia.

6.3.8: Historia De Usuario #18: Asociar Una Asignatura A Una Carrera

Al momento de crear una asignatura, se puede asociar directamente a un programa o a una carrera. Se realiza una conexión foránea con el modelo de Carrera que permite esta vinculación. En la Figura #19, se visualiza el formulario para actualizar una asignatura y el programa a la cual pertenece; si la asignatura ya fue inscrita por un estudiante, no se podrá cambiar de programa o de carrera.

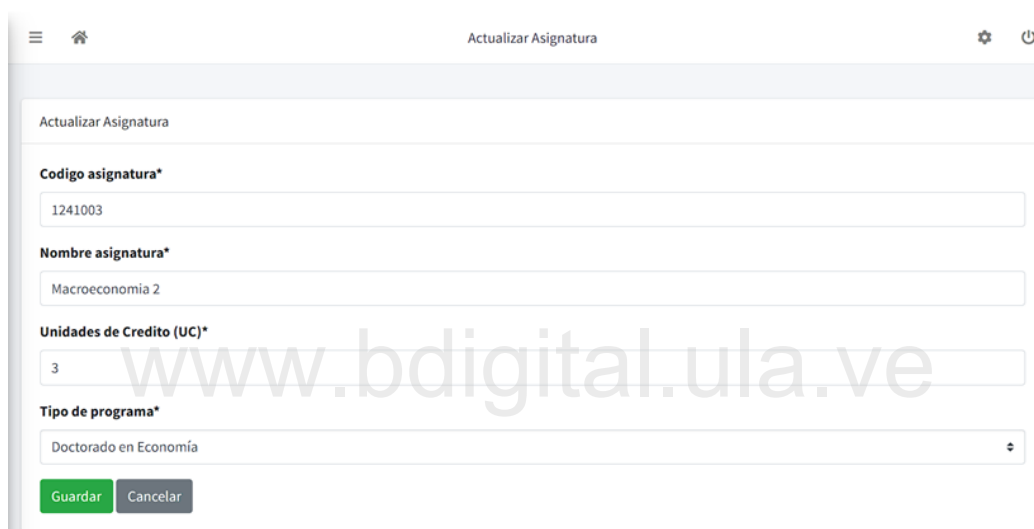


Figura #19. Formulario para actualizar una asignatura. Fuente: Propia.

6.3.9: Historia De Usuario #19: Crear Un Plan De Estudio

Los planes de estudio se utilizarán con el propósito de identificar las menciones de un programa o de una carrera; así, se pueden asociar asignaturas comunes y por especialidad. En la Figura #20, se visualiza el formulario para crear un plan de estudio.

Figura #20. Formulario para crear un plan de estudio. Fuente: Propia.

6.3.10: Historia De Usuario #20: Actualizar Un Plan De Estudio

En la Figura #21, se muestra el formulario para actualizar un plan de estudio seleccionado. El nombre del plan de estudio es único, por lo que, al momento de agregar uno existente, la aplicación web arrojará una notificación de error.

Figura #21. Formulario para actualizar un plan de estudio. Fuente: Propia.

6.3.11: Historia De Usuario #21: Asignar Una Asignatura A Un Semestre Y A Un Plan De Estudio

En la Figura #22, se observa la interfaz del plan de estudio, en la que existe un botón para asociar asignaturas al plan visualizado. Para asociar asignaturas, la Figura #23 muestra el formulario que se encargará de asociar una asignatura seleccionada a un plan de estudio específico y la etapa del semestre en la que se encuentra contemplada.

Detalles de Plan de Estudio

Plan de Estudio

[< Atrás](#)

Nombre	Programa	Descripción	Asociar Asignaturas	Acciones
Maestría Política Económica	Maestría en Economía		+	✎ 🗑

Listado de asignaturas asociadas

Codigo	Asignatura	UC	Etapas del semestre	Fecha de actualización reciente	Acciones
C001	Calculo 01	2	Primer semestre	28 de mayo de 2024 a las 10:42	✎ 🗑
C002	Calculo 02	2	Segundo semestre	28 de mayo de 2024 a las 10:42	✎ 🗑

Figura #22. Interfaz de plan de estudio y de las asignaturas que tiene asociada. Fuente: Propia.

Figura #23. Formulario para asociar una asignatura a un plan de estudio. Fuente: Propia.

6.3.12: Historia De Usuario #22: Actualizar Asignatura Asociada A Un Plan De Estudio

Debido a lógicas de negocio, solo se puede editar la etapa del semestre más no la relación con el plan de estudio. En la Figura #24, se muestra la actualización de la etapa del semestre de una asignatura seleccionada.

Figura #24. Edición de la etapa del semestre de una asignatura en un plan de estudio. Fuente: Propia.

6.3.13: Historia De Usuario #23: Matricular Un estudiante

Para matricular un estudiante, debe utilizar el buscador de estudiante visualizado en la Figura #8 antes, en donde, al introducir la C.I./Pasaporte, arrojará los resultados de los estudiantes que cumplan con el resultado de búsqueda. Al ingresar en el estudiante encontrado, en la Figura #25, se muestra el submenú para ver distintas asociaciones que puede tener el estudiante con las carreras, las facturas, entre otros.



Figura #25. Submenú dentro del módulo de un estudiante existente. Fuente: Propia.

Se debe seleccionar la opción Datos Académicos para visualizar si el estudiante está o no matriculado a algún plan de estudio. De no estar matriculado, se muestra un mensaje con un botón para matricular. En la Figura #26, se visualiza esta interfaz.



Figura #26. Interfaz para matricular un estudiante por primera vez. Fuente: Propia.

El formulario para matricular un estudiante aparecerá luego de hacer clic en el botón Matricular Estudiante. En la Figura #27, se visualiza dicho formulario.

Figura #27. Formulario para matricular a un estudiante en un plan de estudio. Fuente: Propia.

6.3.14: Historia De Usuario #24: Actualizar El Estatus Del Estudiante En Su Matriculación

Para poder editar la matriculación del estudiante, tiene que haber estado ya matriculado. No se puede cambiar el plan de estudio matriculado por reglas de negocio, aunque sí se puede actualizar fecha de matriculación, estatus y cohorte a la que está asignado, tal y como se observa en la figura #28.

Figura #28. Formulario para actualizar la matriculación de un estudiante. Fuente: Propia.

6.4 Sprint N.º 4

En este sprint, se desarrolló toda la experiencia del módulo de registro. Este módulo permite trabajar las inscripciones en un semestre activo para los estudiantes ya matriculados en un plan de estudio y registrar las calificaciones de las asignaturas inscritas. Adicionalmente, se incluye la visualización en el módulo del estudiante para tener acceso directo y fácil sobre sus datos académicos. En la tabla #17, se detallan las historias de usuario trabajadas en este sprint.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
4	2 semanas	#25: Registrar la inscripción de un estudiante en una materia	Registro	3 días
		#26: Actualizar la inscripción de un estudiante	Registro	2 días
		#27: Visualizar todas las inscripciones de un estudiante	Estudiantes	1 día
		#28: Registrar calificación de una asignatura	Registro	3 días
		#29: Actualizar una calificación existente	Registro	2 días
		#30: Visualizar el historial de calificaciones de un estudiante	Estudiantes	1 día

Tabla #17. Historias de usuario incorporadas en el Sprint N.º 4. Fuente: Propia.

6.4.1: Historia de Usuario #25: Registrar La Inscripción De Un Estudiante En Una Materia

Se realizó una lógica para listar en el campo Materia a inscribir las asignaturas que no ha inscrito el estudiante aún, excluyendo cualquier asignatura aprobada o con equivalencia registrada en semestres anteriores, y las asignaturas inscritas en el semestre actual. En la Figura #29, se visualiza el formulario para inscribir una asignatura a un estudiante en un semestre activo.

Figura #29. Formulario para inscribir una asignatura en un semestre a un estudiante. Fuente: Propia.

6.4.2: Historia De Usuario #26: Actualizar La Inscripción De Un Estudiante

Se puede actualizar la inscripción siempre y cuando el semestre en el que se inscribió la asignatura se encuentre activo, al igual que asignaturas que aún no ha inscrito el estudiante, excluyendo cualquier asignatura aprobada o con equivalencia registrada en semestres anteriores y las asignaturas inscritas en el semestre actual. En la Figura #30, se visualiza el formulario para actualizar la inscripción de una asignatura de un estudiante en un semestre activo.

Figura #30. Formulario para actualizar la inscripción de un estudiante en un semestre activo. Fuente: Propia.

6.4.3: Historia De Usuario #27: Visualizar Todas Las Inscripciones De Un Estudiante

Se pueden visualizar las inscripciones inscritas en un semestre activo para el estudiante desde el módulo de Registros o, directamente, en el módulo del estudiante. En la Figura #31, se detallan las asignaturas inscritas por un estudiante en un semestre activo.

Asignaturas inscritas en el plan de estudio Maestría Política Económica

Estudiante: Abraham Mousalli
 CI o Pasaporte: 25151141
 Plan de estudio: Maestría Política Económica

[← Atras](#) [Inscribir Asignaturas +](#)

Asignaturas inscritas en el semestre B2024

	Codigo	Asignatura	UC	Etapas	Acciones
1	C001	Calculo 01	2	Primer semestre	
2	C002	Calculo 02	2	Segundo semestre	

Figura #31. Interfaz para visualizar las asignaturas inscritas en un semestre activo. Fuente: Propia.

6.4.4: Historia De Usuario #28: Registrar Calificación De Una Asignatura

Al momento de registrar una calificación de las asignaturas previamente inscritas, aparecerán listadas dichas asignaturas. Se puede registrar una calificación en el botón con símbolo + verde visualizado en la Figura #32. Al momento de hacer clic en este botón, aparece el formulario para registrar una calificación; dicho formulario se detalla en la Figura #33.

Registro de notas en semestre B2024

Estudiante: Abraham Mousalli
 CI o Pasaporte: 25151141
 Plan de estudio: Maestría Política Económica

Lista de asignaturas inscritas en el semestre B2024

[← Atras](#)

	Codigo	Asignatura	UC	Etapas	Calificacion numerica	Calificacion en letra	Estatus	Acciones
1	C001	Calculo 01	2	Primer semestre	16	Dieciseis	Equivalencia	
2	C002	Calculo 02	2	Segundo semestre	14	Catorce	Aprobado	
3	C003	Calculo 03	3	Tercer semestre	-Sin calificar-	-Sin calificar-	-	

Figura #32. Interfaz para visualizar las asignaturas inscritas y para poder asignar una calificación. Fuente: Propia.

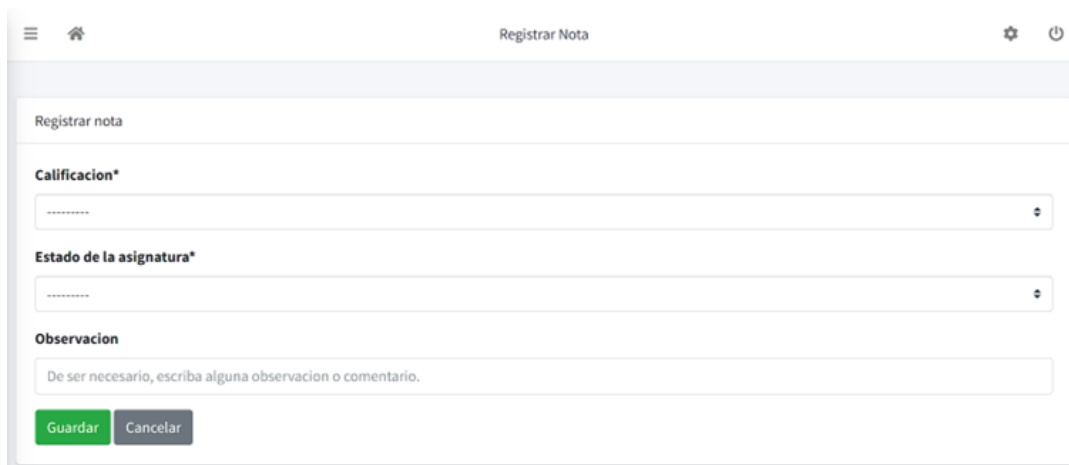


Figura #33. Formulario para registrar una calificación a una asignatura inscrita. Fuente: Propia.

6.4.5: Historia De Usuario #29: Actualizar Una Calificación Existente

Si el semestre en que se inscribió la asignatura sigue activo en la aplicación, se podrá actualizar y modificar ese registro. En la Figura #34, se visualiza el formulario para actualizar la calificación, el estado de la asignatura y la observación.

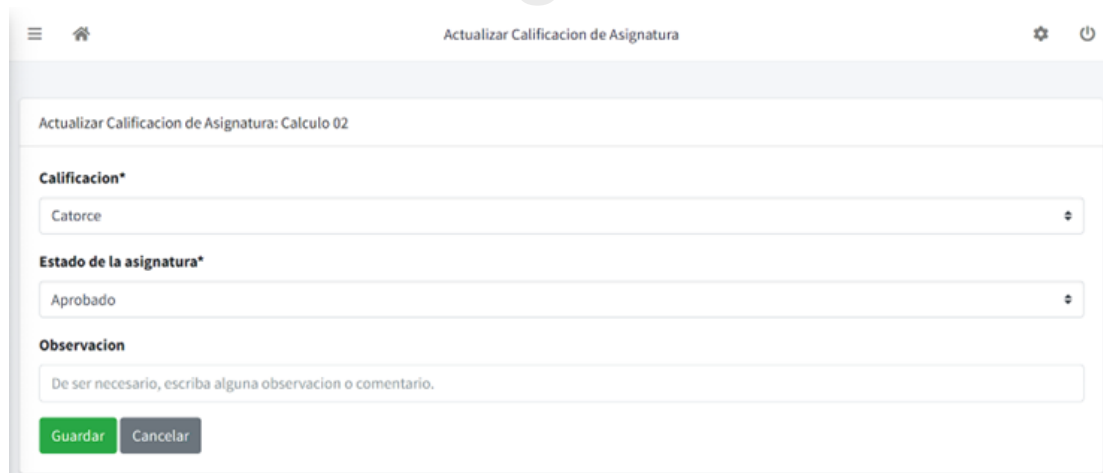


Figura #34. Formulario para actualizar la calificación a una asignatura inscrita. Fuente: Propia.

6.4.6: Historia De Usuario #30: Visualizar El Historial De Calificaciones De Un Estudiante

En el módulo del estudiante, se podrán detallar las asignaturas ya calificadas. Esto permite llevar un control y un registro de las asignaturas cursadas y de sus calificaciones. En la Figura #35, se muestra la visualización de las calificaciones registradas para un estudiante.

Calificaciones

Lista de asignaturas cursadas por semestre

Descargar PDF

Semestre	Codigo	Asignatura	UC	Etap	Calificacion numerica	Calificacion en letra	Estatus	
1	A2024	C003	Calculo 03	3	Tercer semestre	--	Ausente	
2	A2024	E001	Econometria 1	2	Primer semestre	13	Trece	Aprobada
3	B2024	C001	Calculo 01	2	Primer semestre	16	Dieciseis	Equivalencia
4	B2024	C002	Calculo 02	2	Segundo semestre	14	Catorce	Aprobada

Figura #35. Formulario para actualizar la calificación a una asignatura inscrita. Fuente: Propia.

6.5 Sprint N.º 5

En este Sprint, se desarrolló toda la experiencia del módulo de productos. Este módulo permite trabajar los tipos de productos que ofrece actualmente la SEPEC: aranceles, cursos y unidad curricular. Además, se realizó la asociación con la nacionalidad para poder registrar precios basados en diferentes tipos de nacionalidades. Esto sirvió para el módulo de facturas que se desarrolló previamente. En la Tabla #18, se detallan las historias de usuario incluidas en este sprint.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
5	2 semanas	#31: Registrar un nuevo arancel / curso / unidad curricular	Productos	0,5 días
		#32: Actualizar un nuevo arancel / curso / unidad curricular	Productos	0,5 días
		#33: Registrar el precio de un arancel para una nacionalidad específica	Productos	0,5 días
		#34: Actualizar el precio de un arancel para una nacionalidad específica	Productos	0,5 días
		#35: Registrar el precio de un curso para una nacionalidad específica	Productos	0,5 días
		#36: Actualizar el precio de un curso para una nacionalidad específica	Productos	0,5 días
		#37: Registrar el precio de una unidad curricular para una nacionalidad específica	Productos	1 semana
		#38: Actualizar el precio de una unidad curricular para una nacionalidad específica	Productos	3 días

Tabla #18. Historias de usuario incorporadas en el Sprint N.º 5. Fuente: Propia.

6.5.1: Historia De Usuario #31: Registrar Un Nuevo Arancel, Un Curso O Una Unidad Curricular

Se crea la misma experiencia para registrar los 3 tipos de productos. En la Figura #36, se visualiza el formulario para crear un nuevo arancel, con la condición de que el nombre debe ser único. Esta experiencia se repite para curso y para unidad curricular.

Figura #36. Formulario para crear un arancel. Fuente: Propia.

6.5.2: Historia De Usuario #32: Actualizar Un arancel, Un curso O Una Unidad Curricular

En la Figura #37, se visualiza el formulario para actualizar un arancel; el nombre debe ser único, por lo que, si intenta actualizar y no se cumple esta condición, la aplicación arrojará una notificación de error. Esta experiencia se repite para curso y para unidad curricular.



Figura #37. Formulario para actualizar un arancel. Fuente: Propia.

6.5.3: Historia De Usuario #33: Registrar El Precio De Un Arancel Para Una Nacionalidad Específica

Es necesario registrar precios por nacionalidades específicas, ya que esto permitirá facturar a los estudiantes de acuerdo a su nacionalidad posteriormente. En la Figura #38, se visualiza el formulario para crear el precio de un arancel para una nacionalidad específica.

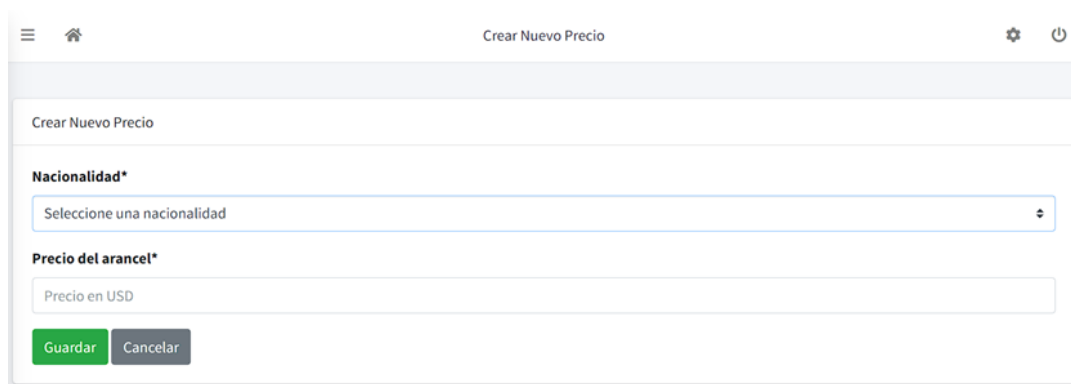


Figura #38. Formulario para crear el precio de un arancel para una nacionalidad específica. Fuente: Propia.

Esta misma experiencia y formulario se repite para las historias de usuario #35 y #37, solo que, en vez de a un arancel, se asocia a un curso o a un tipo de unidad curricular.

6.5.4: Historia De Usuario #34: Actualizar El Precio De Un Arancel Para Una Nacionalidad Específica

Es necesario poder controlar los precios de los productos, ya que, para un futuro, se podrá facturar el precio real y no el existente. En la Figura #39, se visualiza el formulario para actualizar el precio de un arancel para una nacionalidad específica; debido a una regla de negocio, no se puede cambiar el tipo de nacionalidad previamente registrada.

The image shows a web application interface for updating a price. At the top, there is a header bar with a menu icon, a home icon, the title 'Actualizar Precio', and settings and power icons. Below the header, a form is displayed with the title 'Actualizar precio asociado al Arancel: Constancia de notas'. Inside the form, there is a label 'Nacionalidad: Venezolano' and a label 'Precio del arancel*'. Below the 'Precio del arancel*' label is a text input field containing the number '5'. At the bottom of the form are two buttons: 'Guardar' (green) and 'Cancelar' (gray). A large, semi-transparent watermark 'www.bdigital.ula.ve' is overlaid across the center of the form.

Figura #39. Formulario para actualizar el precio de un arancel para una nacionalidad específica. Fuente: Propia.

Esta misma experiencia y formulario se repite para las historias de usuario #36 y #38, solo que, en vez de un arancel, se asocia a un curso o a un tipo de unidad curricular.

6.6 Sprint N.º 6

En este Sprint, se desarrolló toda la experiencia del módulo de facturas. Este módulo permite facturar a los estudiantes diferentes productos que ofrece actualmente la SEPEC, desde aranceles, cursos y precio por unidad curricular. Se puede facturar asignaturas inscritas, aplicar

descuentos, incluir varios tipos de productos, registrar precios y obtener el saldo o la deuda de un estudiante. En la Tabla #19, se detallan las historias de usuario incluidas en este sprint.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
6	2 semanas	#39: Crear una nueva factura para un estudiante	Facturas	2 días
		#40: Agregar un elemento a una factura existente	Facturas	1 semana
		#41: Registrar un pago para una factura	Facturas	1 semana

Tabla #19. Historias de usuario incorporadas en el Sprint N.º 6. Fuente: Propia.

6.6.1: Historia De Usuario #39: Crear Una Nueva Factura Para Un Estudiante

Al momento de crear la factura, solo se solicita la fecha de vencimiento, un estado de factura, la forma de pago y un comentario de ser necesario; el resto de los datos, como lo son subtotal, saldo, descuento y monto, no se rellenan. La razón es que, al momento de crear la factura, no hay productos asociados, por lo que, hasta que no se agreguen, esos datos estarán nulos. En la Figura #40, se visualiza el formulario para crear una nueva factura asociada a un estudiante.

Crear una factura

Crear Nueva Factura

Fecha de vencimiento*

dd/mm/aaaa

Estado de la factura*

Pendiente

Forma de pago*

Contado

Descuento

Comentario

De ser necesario, escriba alguna observacion o comentario.

Guardar Cancelar

Figura #40. Formulario para crear una nueva factura a un estudiante. Fuente: Propia.

6.6.2: Historia De Usuario #40: Agregar Un Elemento A Una Factura Existente

Para agregar elementos a una factura, basta con ingresar en ella y hacer clic en el botón Añadir Producto. En la Figura #41, se muestra la interfaz de la factura y en donde ubicar el formulario para incluir productos.

The screenshot shows a web interface for viewing invoice details. At the top, there's a header with a menu icon, a home icon, the title 'Detalles de Factura', and settings and power icons. Below the header, there's a navigation bar with a '< Atrás' button and an edit icon. The main content area is titled 'Factura #00013' with a subtitle 'Fecha de creacion: 02/Jun/2024'. It is divided into two columns: 'Detalles del estudiante' and 'Estatus de la factura'. The student details include CI/Pasaporte: 25151141, Nombre: Abraham, Apellidos: Mousalli, and Nacionalidad: EXR. The invoice status includes Fecha de vencimiento: 31/May/2024, Estatus: Pendiente (in a red box), Condicion de pago: Contado (in a green box), and Ultima actualizacion: 02/Jun/2024. Below this, there's a section titled 'Aun no se ha agregado ningun producto' with a message 'Debes agregar productos a la factura.' and a green 'Añadir Producto' button. At the bottom, there's a section titled 'Detalles de pagos registrados' with a message 'Aun no hay productos en la factura' and 'Esta factura aun no tiene agregado algun producto.' followed by 'Agregue productos a la factura para poder registrar pagos.'

Figura #41 Interfaz de una factura asociada a un estudiante. Fuente: Propia.

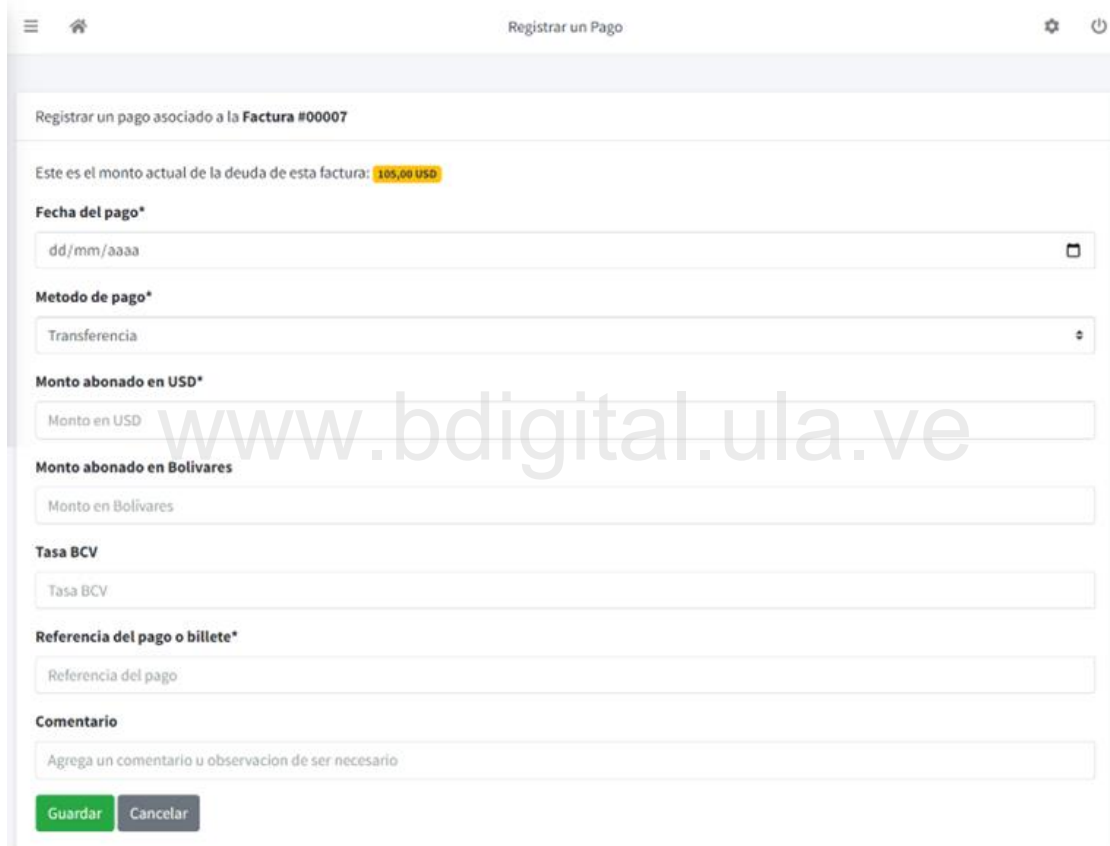
En la Figura #42, se visualiza el formulario para agregar un producto a una factura. Se necesita seleccionar el tipo de producto para que puedan desplegarse, de manera dinámica, los productos creados. Ya vienen filtrados con el precio según la nacionalidad del estudiante.

The screenshot shows a web interface for adding a new element to an invoice. The header has a menu icon, a home icon, the title 'Añadir un Elemento a una Factura', and settings and power icons. The main content area is titled 'Añade un nuevo elemento a la factura'. It contains a 'Descuento' field and a 'Tipo de Producto*' dropdown menu. The dropdown menu is open, showing options: 'Arancel', 'Curso', and 'Unidad Curricular'. At the bottom, there are 'Guardar' and 'Cancelar' buttons.

Figura #42. Formulario para facturar un producto a un estudiante. Fuente: Propia.

6.6.3: Historia De Usuario #41: Registrar Un Pago Para Una Factura

Al momento de abonar o DE cancelar una factura de un estudiante, se debe registrar un pago. En la Figura #43, se muestra el formulario para registrar un pago a una factura de un estudiante, que incluye una visualización de la deuda que posee el estudiante según los productos facturados.



Registar un Pago

Registar un pago asociado a la Factura #00007

Este es el monto actual de la deuda de esta factura: 105,00 USD

Fecha del pago*

dd/mm/aaaa

Método de pago*

Transferencia

Monto abonado en USD*

Monto en USD

Monto abonado en Bolívars

Monto en Bolívars

Tasa BCV

Tasa BCV

Referencia del pago o billete*

Referencia del pago

Comentario

Agrega un comentario u observación de ser necesario

Guardar Cancelar

Figura #43. Formulario para registrar un pago a la factura de un estudiante. Fuente: Propia.

6.7 Sprint N.º 7

Para este último Sprint, se desarrolló toda la experiencia del módulo de reportes. Este módulo ayuda a obtener información importante sobre la situación académica de los estudiantes,

las notas registradas en las asignaturas inscritas en distintos semestres y la situación financiera o administrativa. En la Tabla #20, se detallan las historias de usuario incluidas en este sprint.

Sprint Backlog				
Sprint N°	Duración del Sprint	Historia de Usuario	Modulo	Tiempo de Desarrollo
7	1 semana	#42: Generar un reporte de calificaciones de un estudiante por semestre	Reportes	2 días
		#43: Generar un reporte de pagos efectuados en un mes específico	Reportes	1 día
		#44: Generar un reporte de estudiantes matriculados por carrera	Reportes	1 día
		#45: Generar un reporte de todas las facturas	Reportes	1 día

Tabla #20. Historias de usuario incorporadas en el Sprint N.º 7. Fuente: Propia.

6.7.1: Historia De Usuario #42: Generar Un Reporte De Calificaciones De Un Estudiante Por Semestre

Para generar este reporte, es necesario ingresar en el módulo del estudiante al que se desea obtener dicho reporte; el estudiante debe contar con, al menos, una asignatura inscrita y con calificación para poder generarlo. En la Figura #44, se visualiza el PDF que se genera con las notas obtenidas en las asignaturas inscritas en distintos semestres.

Estudiante: Abraham Mousalli
 CI/Pasaporte: 25151141
 Programa: Maestría en Economía. Fecha de Matriculación: 28-05-2024
 Plan de Estudio: Maestría Política Económica
 Cohorte: 2024

Semestre: A2024

Fecha de Inicio: 01-01-2024. Fecha de Finalización: 10-05-2024

Codigo	Asignatura	UC	Etapas	Calificacion numerica	Calificacion en letra	Estatus
C003	Calculo 03	3	Tercer semestre	--	--	Ausente
E001	Econometria 1	2	Primer semestre	13	Trece	Aprobado

Semestre: B2024

Fecha de Inicio: 28-05-2024. Fecha de Finalización: 31-12-2024

Codigo	Asignatura	UC	Etapas	Calificacion numerica	Calificacion en letra	Estatus
C001	Calculo 01	2	Primer semestre	16	Dieciseis	Equivalencia
C002	Calculo 02	2	Segundo semestre	14	Catorce	Aprobado

Figura #44. Reporte PDF de calificaciones registradas a un estudiante por semestre. Fuente: Propia.

6.7.2: Historia De Usuario #43: Generar Un Reporte De Pagos Efectuados En Un Mes Específico

En la Figura #45, se puede apreciar el reporte generado en formato PDF para detallar los pagos recibidos y registrados en facturas en un mes y en un año previamente seleccionados.

# Factura	CI/Pasaporte	Nombre	Apellido	Fecha del pago	Metodo	Monto USD	Monto VES	Referencia
00001	25151141	Abraham	Mousalli	28/05/2024	Transferencia	15.00		A202425
00003	9471742	Albio	Marquez	28/05/2024	Efectivo	100.00		A202425
00003	9471742	Albio	Marquez	29/05/2024	Transferencia	5.00	200.00	123456789
00004	9471742	Albio	Marquez	29/05/2024	Transferencia	165.00		A202425
00004	9471742	Albio	Marquez	29/05/2024	Transferencia	20.00		A202425
00002	25151141	Abraham	Mousalli	29/05/2024	Transferencia	100.00		A202425
00002	25151141	Abraham	Mousalli	29/05/2024	Efectivo	15.00		A202425
00006	25151141	Abraham	Mousalli	30/05/2024	Transferencia	100.00		A202425
00010	9471742	Albio	Marquez	31/05/2024	Transferencia	0.00		A202425
00010	9471742	Albio	Marquez	30/05/2024	Transferencia	195.00	4000.00	123456
00012	9471742	Albio	Marquez	31/05/2024	Transferencia	25.00		A202425
Totales						740.00	4200.00	

Figura #45. Reporte PDF para listar los pagos recibidos para un mes/año seleccionado. Fuente: Propia.

6.7.3: Historia De Usuario #44: Generar Un Reporte De Estudiantes Matriculados Por Carrera

En la Figura #46, se visualiza el PDF generado en donde se listan todos los estudiantes matriculados a un plan de estudio, junto con datos de cada estudiante registrado y la cohorte a la que pertenece.

CI/Pasaporte	Nombre	Apellido	Programa	Estatus	Cohorte	Fecha de inicio	Fecha de posible culminacion
25151141	Abraham	Mousalli	Maestria en Economia	Activo	2024	27/05/2024	26/05/2029
25152612	Gabriel Andres	Belandria Lopez	Doctorado en Economia	Activo	2024	27/05/2024	26/05/2030
9471742	Albio	Marquez	Doctorado en Economia	Activo	2024	27/05/2024	26/05/2030
9471742	Albio	Marquez	Maestria en Economia	Activo	2024	27/05/2024	26/05/2029

Figura #46. Reporte PDF de la lista de estudiantes matriculados a un plan de estudio. Fuente: Propia.

6.7.4: Historia De Usuario #45: Generar Un Reporte De Todas Las Facturas

En la Figura #47, se puede apreciar el reporte generado en formato Excel para obtener todas las facturas generadas en la aplicación web. Esto permitirá tener un control del estatus de las facturas para el control administrativo.

# Factura	Cl/Pasaporte	Nombre	Apellido	Fecha de vencimiento	Estatus	Condicion de Pago	Monto a Factura	Saldo/Deuda USD
00001	25151141	Abraham	Mousalli	30/05/2024	Cancelada	Contado	15	0
00002	25151141	Abraham	Mousalli	28/05/2024	Cancelada	Contado	115	0
00003	9471742	Albio	Marquez	30/05/2024	Cancelada	Contado	105	0
00004	9471742	Albio	Marquez	29/05/2024	Cancelada	Contado	185	0
00005	25151141	Abraham	Mousalli	30/06/2024	Pendiente	Credito	150	150
00006	25151141	Abraham	Mousalli	30/05/2024	Cancelada	Contado	100	0
00007	25151141	Abraham	Mousalli	31/05/2024	Pendiente	Contado	105	105
00008	25151141	Abraham	Mousalli	30/06/2024	Pendiente	Contado		
00009	25152612	Gabriel An	Belandria	30/05/2024	Pendiente	Contado		
00010	9471742	Albio	Marquez	30/06/2024	Cancelada	Contado	195	0
00011	25151141	Abraham	Mousalli	31/05/2024	Pendiente	Contado	150	25
00012	9471742	Albio	Marquez	31/05/2024	Pendiente	Contado	45	20
00013	25151141	Abraham	Mousalli	31/05/2024	Pendiente	Contado		

Figura #47. Reporte Excel de las facturas creadas. Fuente: Propia.

www.bdigital.ula.ve

Capítulo VII

Conclusiones y Recomendaciones

7.1 Conclusiones

La digitalización de la información es crucial para cualquier entidad educativa, ya que agiliza el tiempo de consulta y mejora la toma de decisiones. Gracias a la implementación del *framework* Django, que ofrece la posibilidad de trabajar en distintos ámbitos y en roles de la ingeniería, el desarrollo de la aplicación web se llevó a cabo a través de una serie de sprints bien definidos. Esto permitió una estructuración ordenada y eficiente, asegurando la consecución de los objetivos planteados.

El módulo para inscribir asignaturas y registrar calificaciones proporcionó el seguimiento necesario sobre el rendimiento académico de los estudiantes; además, cuenta con acceso sencillo y práctico, centralizando la información de cada estudiante.

La funcionalidad para crear aranceles, cursos y unidades curriculares fue implementada con éxito, permitiendo asociar un precio a cada producto según la nacionalidad del estudiante. Esto beneficia el registro y el control de la gestión administrativa, y permite generar reportes que mejorarán, significativamente, la operatividad de la SEPEC.

Las validaciones de datos y las notificaciones ayudan a mostrar, en tiempo real, cualquier error lógico o de usuario, evitando inconsistencias en la información almacenada.

Cada sprint abordó un componente crucial del sistema, permitiendo un desarrollo modular y manejable. El enfoque ágil facilitó la adaptación a los cambios y la mejora continua, asegurando que el proyecto cumpliera con los requisitos y expectativas establecidos.

7.2 Recomendaciones

La aplicación web es altamente escalable y puede ser implementada en distintas instituciones educativas. Se recomienda:

- Publicar la aplicación web en un servicio en la nube o de manera local para uso interno.
- Modularizar los *templates* y los ficheros, y mejorar aspectos de usabilidad y de rendimiento.
- Continuar el desarrollo de reportes personalizados.
- Incluir un módulo de configuración de cuenta y otro módulo de registro de usuarios.
- Convertir este proyecto en una solución *white-label* para que cualquier institución educativa pueda personalizarla con su propia marca, logotipo y diseño sin necesidad de desarrollar la tecnología desde 0.

Estas recomendaciones están basadas en la experiencia adquirida durante el desarrollo del proyecto y proporcionan una guía práctica para futuros desarrollos, asegurando la eficiencia, la calidad y la sostenibilidad de los sistemas educativos y administrativos.

Referencias Bibliográficas

- Alvarez Miranda, V. (2007). Desarrollo de un Sistema de Inscripciones utilizando AJAX (Tesis de licenciatura). Universidad de las Américas Puebla, Departamento de Computación, Electrónica y Mecatrónica, Escuela de Ingeniería y Ciencias.
- Arias. (2012). El Proyecto de Investigación (6.^a ed.). Caracas, Venezuela: Episteme.
- Arnold, M. y Osorio, F. (1998). Introducción a los conceptos básicos de la teoría general de sistemas. Cinta moebio, 3, 40-49.
- Averruz, H.; Sevilla, J. y Martínez, G. (2019). Psicología Actual, la modernización de la ciencia, vol. 6, núm. 1 [archivo PDF]. Recuperado en 22 de enero de 2024, de: <https://www.redalyc.org/journal/5395/539557206030/539557206030.pdf>.
- Cadavid, A.; Martínez, J., y Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva, 11(2), 30-39.
- Costa, D. (2002). El modelo relacional y el álgebra relacional. UOC, la universidad virtual.
- Dubs, Renie (2003). Una estrategia metodológica para el proyecto factible. Entretemas, Año 1 [archivo PDF]. Recuperado en 22 de enero de 2024, de: <https://n9.cl/sxfqn>.
- Flores, W. (2022). Sistema de información para la gestión de eventos académicos en las diferentes instituciones educativas de nivel superior en Moquegua. Trabajo de grado. Universidad Nacional De Moquegua, Perú.
- García, D.; Argueta, W. y Hernández, E. (1997). Automatización Del Proceso De Registro Académico Para La Universidad Don Bosco. Proyecto de grado publicado. Recuperado en 23 de enero de 2024, de: <https://n9.cl/1okxtj>.
- Gómez, J. (2012) Evolución Histórica De Los Lenguajes De Programación. Un viaje a la historia de la Informática, 27.
- González, R. (2010). Arquitectura web. Universitat de Lleida, España.
- Henao, M., y Arango, M. (2006). Soluciones tecnológicas que apoyan la gestión del conocimiento. AD-minister, (8), 69-85. Universidad EAFIT, Medellín, Colombia.
- Hernández, A. (2003). Los sistemas de información, evolución y desarrollo. Proyecto social: Revista de relaciones laborales, 10-11, 149-165. ISSN 1133-3189.
- Huete Matus, M., y Carrillo Blandón, G. (2017). Automatización de los procesos de inscripción y divulgación de la JUDC. Universidad Y Ciencia, 7(11). Recuperado en 30 de enero de 2024, de: <https://doi.org/10.5377/uyc.v7i11.4442>.

- Hurtado, J. (2008). Metodología Holística. Caracas. Editorial Fundación Sypal. Tercera Edición.
- Inciarte, A.; Marcano, N. y Reyes, M. (2006). Gestión académico-administrativa en la educación básica. Revista Venezolana de Gerencia, 11(34), 221-243. Recuperado en 31 de enero de 2024, de: <https://n9.cl/oxr7q>.
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2010). Database concepts (pp. 1480-1486). Upper Saddle River, NJ: Prentice Hall.
- Lijeron, D. (2009). Sistema Integrado De Inscripción, Seguimiento, Biblioteca Y Caja. Caso: Renovación Carismática Católica La Paz Zona Central. Proyecto de grado publicado. Universidad Mayor De San Andrés, Bolivia.
- López Navarro, F. (2021). La Digitalización como herramienta para la Educación Financiera [archivo PDF]. Recuperado en 05 de febrero de 2024, de: <https://www.researchgate.net/publication/357028068>.
- Luján-Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.
- Macías, V. y Mindiola, J. (2018). Desarrollo de un sistema de gestión de contenidos web utilizando metodología Scrum para Educación Continua-ESPOL. Trabajo de grado publicado. Escuela Superior Politécnica Del Litoral, Ecuador.
- Martínez, M., y Vivas, A. (2022). Guía de Modalidad de Proyecto Factible: etapas, propuesta, ejecución y evaluación. Santiago de Chile: Universidad Miguel de Cervantes, Programa de Licenciatura en Educación.
- Maida, E. y Pacienza, J. (2015). Metodologías de desarrollo de software.
- Mendoza, S. y Avila, D. (2020). Técnicas e instrumentos de recolección de datos. Boletín científico de las ciencias económico administrativas del ICEA, 9(17), 51-53.
- Miranda, V. (2007). Desarrollo de un Sistema de Inscripciones utilizando AJAX. Trabajo de grado. Universidad de las Américas, Puebla, México.
- Project Management. (2023, octubre). The Scrum framework. LinkedIn. Recuperado en 07 de febrero de 2024, de: <https://n9.cl/w7vit>.
- Rad, N. K., y Turley, F. (2019). Los fundamentos de agile Scrum. Van Haren.
- Reyes, P. y Marín. R. (2021). Aplicación web empleando la metodología xp para la gestión académica del Instituto de Informática de la Universidad Nacional del Altiplano Puno - 2019. Trabajo de grado. Universidad Nacional Del Altiplano, Perú.

- Rodríguez, C. y Dorado, R. (2015). ¿Por qué implementar Scrum? Revista ONTARE, Vol. 3, N.º 1, (Ejemplar dedicado a: Aplicaciones en Ingeniería), págs. 125-144. Recuperado en 10 de febrero de 2024, de: <https://dialnet.unirioja.es/servlet/articulo?codigo=8705520>.
- Salazar, L. (2021). La Universidad innovadora. Una propuesta para la universidad venezolana.
- Schwaber, K., y Sutherland, J. (2020). The Scrum Guide. Scrum. Org. Recuperado en 10 de febrero de 2024, de: <http://www.scrum.org/resources/scrum-guide>.
- Tamayo, M. (2003). El proceso de la Investigación Científica (4.ª ed.). Limusa. Recuperado de 15 de febrero de 2024, de: https://www.enfermeriaaps.com/portal/?wpfb_dl=4387.
- Ulloa Jaramillo, J. J. (2014-12). Tesis. Recuperado en 27 de enero de 2024, de: <http://repositorio.ug.edu.ec/handle/redug/6511>.
- UNIR Revista. (2022). ¿Qué es un framework en el desarrollo de software? UNIR Revista. Recuperado en 08 de febrero de 2024, de: <https://n9.cl/rfaap>.
- Universidad Pedagógica Experimental Libertador (2003). Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales. Caracas: Autor.
- Urbano, J. G., y Gómez, J. C. (2022). Desarrollo e implementación de un sistema de información web para el programa federal de la SEP “RENOES”(rol administra-dor). Revista Ingeniantes, 9(2), 1.