



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

Sistema Web de Análisis de Datos para
identificar las causas de deserción en la carrera
de Ingeniería en Tecnologías de la Información
de la BUAP

TESIS

Para obtener el título de
Ingeniera en Tecnologías de la Información

Presenta

Mónica Portillo Sánchez

Asesor

M.I. Carlos Armando Ríos Acevedo



Puebla, Pue. Agosto 2022

Antes de irme...

“El cambio más complicado pero el más satisfactorio, es el que se hace con uno mismo, el ser humano es tan firme en su paso que no necesita una marca física para saber que estuvo ahí, solo sus obras quedan en la memoria colectiva de unos cuantos...”

-Anónimo

Agradezco a Dios y a la vida la oportunidad de estar presente y poder aprovechar los momentos de aprendizaje que me ha dado.

A mis padres, Rocío y Felipe, por apoyarme y permitir que haya logrado cumplir un sueño, además de ser ejemplo de lo quiero y debo superar en la vida, hacer todo para bien.

A Lidia y Ana Karen por ser mis cómplices en esta aventura y a Martin y Majo, por ser esos motores que me impulsan a no decaer.

A mi maestro, amigo y mentor Carlos Armando Ríos Acevedo, gracias por creer en mi persona, en el talento que vio en mí y por la amistad que me ha brindado.

A Vale, Carlos, Mireya, Carmen, Cari, Sam, Isaías, Monse y todos mis amigos por todas las veces que me han brindado su cariño y su amistad en todo momento para cumplir este sueño.

A la maestra Marcela Zapata León, por ser un apoyo y un ejemplo del impacto de una educadora especial para pequeños con discapacidad junto con todos los profesores que han aportado a mi crecimiento académico.

A Alberto y Adrián, por ser mis amigos, maestros y consejeros en la carrera, todo lo que me han dado y todo lo que me enseñan siempre está conmigo a pesar de la distancia.

A Dylan, Kebyn y Viri por ser esos hermanos que la carrera te da y también por ser esa luz cuando caí en momentos de oscuridad y por cuidarme.

A mis X-Women, gracias por hacerme florecer, salir de mi zona y crecer tanto personal como profesionalmente.

A mis chicos 2019, por aceptarme como su mentora y asesora, crecí y descubrí mi vocación gracias a ustedes.

A mi universidad y a mi facultad, por forjar mi espíritu profesional.

A mi pingüina, por enseñarme a ser más fuerte, te agradezco por todos esos momentos bonitos que me diste la oportunidad de poder compartir contigo, fue la época más bonita de la universidad y de mi vida, deseo que seas feliz desde el fondo de mi corazón.

ÍNDICE

ÍNDICE DE CONTENIDO GRÁFICO	5
INTRODUCCIÓN	7
Capitulo I. Descripción del proyecto	8
1.1 Objetivo general	8
1.2 Objetivos Específicos	8
1.3 Metodología	8
1.4 Estado del arte.....	9
1.5 Resultados esperados	10
Capitulo II. Marco Teórico	10
2.1 SCRUM	10
2.1.1 Equipo Scrum	11
2.1.2 Product Owner	11
2.1.3 Scrum Master.....	11
2.1.4 Development Team	11
2.2 MVC.....	11
2.3 PHP.....	12
2.4 SQL	12
2.5 CodeIgniter	12
2.6 Grocery CRUD.....	12
2.7 Balsamiq Wireframes.....	12
2.8 Figma.....	12
2.9 Enterprise Architect	12
2.10 Chart.js	13
2.11 Node.js	13
2.12 Jest	13
Capitulo III. Análisis y Diseño	13

3.1 Especificación de requerimientos	13
Especificación de Requisitos según el estándar de IEEE 830	13
3.2 Análisis del sistema	21
3.2.1 Casos de Uso	21
3.2.2 Diagrama de clases	27
3.2.3 Interfaces Gráficas.....	29
3.3 Desarrollo SCRUM	37
3.3.1 Rol de usuario e historias de usuario	37
3.3.2 Estimación de Historias de Usuario.....	40
3.3.3 Product Backlog	41
3.3.4 Plan de entregas.....	42
Capitulo IV. Diseño de la base de datos	43
4.1 Diseño lógico	43
4.1.1 Modelo Entidad Relación.....	44
4.1.2 Diccionario de datos.....	45
4.2 Diseño Físico.....	49
4.2.1 Modelo Entidad Relación Extendido	49
4.3 Código SQL	50
Capítulo V. Implementación y pruebas.....	50
5.1 Implementación.....	50
5.1.1 Inicio de Sesión.....	51
5.1.2 Informe principal de carrera.....	51
5.1.3 Cortes Generacionales y Mapas Gráficos.....	52
5.1.4 Reportes por Materia	54
5.1.5 Reportes por Corte generacional.....	57
5.2 Pruebas	58
5.2.1 Pruebas Caja Negra	59

5.2.2 Pruebas Unitarias	62
Capítulo VI. Conclusiones y trabajos futuros	66
6.1 Conclusiones.....	66
6.2 Trabajo futuro.....	66
Referencias	67
ANEXOS	68

ÍNDICE DE CONTENIDO GRÁFICO

Figura 1. Marco de trabajo SCRUM.....	11
Figura 2. Diagrama de Casos de Uso.....	22
Figura 3. Diagrama de Clases	27
Figura 4. Prototipo Inicio de sesión.....	29
Figura 5. Prototipo Ventana Principal	30
Figura 6. Prototipo de sección "Estudiantes"	30
Figura 7. Prototipo de listado de estudiantes de X generación.....	31
Figura 8. Prototipo de Mapa Curricular	31
Figura 9. Prototipo de sección "Reportes"	32
Figura 10. Prototipo de modal para Reporte por Materia.....	32
Figura 11. Prototipo de Reporte por materia - Sección de gráficas	33
Figura 12. Interfaz de alta fidelidad - Inicio de Sesión	34
Figura 13. Interfaz de alta fidelidad - Principal.....	34
Figura 14. Interfaz de alta fidelidad - Estudiantes.....	35
Figura 15. Interfaz de alta fidelidad - Lista de estudiantes.....	35
Figura 16. Interfaz de alta fidelidad - Mapa gráfico sin marcar	36
Figura 17. Interfaz de alta fidelidad - Mapa gráfico marcado	36
Figura 18. Interfaz de alta fidelidad - Menú Reportes	37
Figura 19. Historia de Usuario - Inicio de Sesión.....	38
Figura 20. Historia de Usuario - Visualizar informe general.....	38
Figura 21. Historia de Usuario - Registrar un cardex.....	38
Figura 22. Historia de Usuario - Actualizar un cardex.....	38
Figura 23. Historia de Usuario - Visualizar mapa gráfico.....	39
Figura 24. Historia de Usuario - Consultar generación	39
Figura 25. Historia de Usuario - Buscar estudiante	39
Figura 26. Historia de Usuario - Crear reporte por materia.....	39

Figura 27. Historia de Usuario - Crear reporte por corte generacional	40
Figura 28. Historia de Usuario - Exportar reporte	40
Figura 29. Historia de Usuario - Mostrar gráficas	40
Figura 30. Modelo Entidad Relación.....	44
Figura 31. Modelo Entidad Relación Extendido con Workbench	50
Figura 32. Inicio de Sesión	51
Figura 33. Informe principal	52
Figura 34. Informe principal (parte 2).....	52
Figura 35. Cortes Generacionales	53
Figura 36. Lista de estudiantes de una generación	53
Figura 37. Mapa gráfico de un estudiante	54
Figura 38. Sección mapa gráfico para optativas.....	54
Figura 39. Menú para reportes	55
Figura 40. Reporte por materia.....	55
Figura 41. Reporte de una materia (primera gráfica).....	56
Figura 42. Reporte de una materia (segunda gráfica)	56
Figura 43. Tablas de datos para reporte.....	57
Figura 44. Reporte por corte generacional	57
Figura 45. Reporte por corte generacional (despliegue).....	58
Figura 46. Reporte por corte generacional (vista de gráfica)	58
Figura 47. Clase "piloto" en JavaScript.....	62
Figura 48. Muestra de prueba "piloto" para trabajar con Jest.....	63
Figura 49. Ejecución de prueba piloto con Jest en Node.js	63
Figura 50. Prueba Unitaria para creación de materias	64
Figura 51. Prueba Unitaria de visualización	64
Figura 52. Prueba Unitaria de búsqueda de materia en cardex	65
Figura 53. Prueba Unitaria de Marcado de materias con banderas	65
Tabla 1. Descripción Caso de Uso Autenticarse.	22
Tabla 2. Descripción Caso de Uso Visualizar Historial.	23
Tabla 3. Descripción Caso de Uso Generar Reportes.....	24
Tabla 4. Descripción Caso de Uso Administrar Corte Generacional.	26
Tabla 5. Historias de usuario	41
Tabla 6. Product Backlog.....	41
Tabla 7. Sprint 1	42
Tabla 8. Sprint 2	42
Tabla 9. Sprint 3	43
Tabla 10. Sprint 4	43
Tabla 11. Sprint 5	43
Tabla 12. Sprint 6	43
Tabla 13. DDD de entidad Área.....	45
Tabla 14. DDD de entidad Nivel	45

Tabla 15. DDD de entidad Periodo	45
Tabla 16. DDD de entidad Materia	46
Tabla 17. DDD de entidad Clave	46
Tabla 18. DDD de entidad Alumno	46
Tabla 19. DDD de entidad Materia Aprobada	47
Tabla 20. DDD de entidad Corte Generacional	48
Tabla 21. DDD de entidad Administrador	48
Tabla 22. DDD de entidad Plan Cuatrimestral	49
Tabla 23. DDD de entidad Plan Semestral	49
Tabla 24. Pruebas de caja negra.....	59

INTRODUCCIÓN

La Ingeniería en Tecnologías de la Información es una carrera que presenta un aumento en su matrícula cada año y, esto permite a más jóvenes interesados en la tecnología a través de la innovación constante y un deseo de superación para prepararse dentro de la BUAP. Cuenta actualmente con un poco más de 600 alumnos matriculados en ella.

En cada ciclo escolar entra alrededor de 150 a 200 estudiantes aproximadamente, de los cuáles solo un pequeño rango alcanza a egresar y titularse. Muchos de estos llegan a retrasarse con materias o en el peor de los escenarios, dejan los estudios por distintos factores que afectan interna o externamente a este.

En los últimos años se ha notado una baja considerable dentro de la misma, siendo notorio en la cantidad de egresados y de titulados sea mínima a la cantidad que ingresan, así como la cantidad de alumnos que continúan después de los primeros dos semestres. Los datos de la Asociación Nacional de Universidades e Instituciones de Educación Superior (ANUIES) en su anuario estadístico [1] del periodo escolar 2018-2019 indicó que la población existente era de 562 estudiantes, de los cuales 36 lograron egresar y 10 alcanzaron a titularse.

La deserción escolar ha sido un problema que en los últimos años ha ido en aumento y las dependencias de cada universidad han planeado estrategias para disminuirla para lograr que la mayoría de los estudiantes logren permanecer y, por consiguiente, culminar sus estudios de forma satisfactoria, pero se ha convertido en una tarea compleja que implica un reto muy grande por afrontar.

Para este fin se propuso un sistema web que permita gestionar la información académica de los estudiantes, a través de técnicas de minería de datos e inteligencia de negocios poder identificar patrones dentro de estos que permitan plantear estrategias a través de la generación de proyecciones para

disminuir la deserción en ITI. Los datos que se trabajaron provienen de los historiales académicos que tiene la coordinación correspondiente, usando una muestra del 30% de la población, por lo que el repositorio GitHub donde se almacenó el código y la base de datos se encuentra de forma privada, además de que el software creado es de uso exclusivo del coordinador(a) en turno.

Capitulo I. Descripción del proyecto

1.1 Objetivo general

Desarrollar una aplicación web que permita analizar y visualizar los datos de los estudiantes para determinar las estrategias a seguir con el fin de evitar que abandone la carrera.

1.2 Objetivos Específicos

- Desarrollo de interfaces CRUD (Create, Read, Update and Delete).
- Implementación de un dashboard para el manejo de la información de los estudiantes.
- Analizar los datos obtenidos a partir del seguimiento de la ruta académica de cada estudiante que esté almacenado en la base de datos por medio de algoritmos de minería de datos.

1.3 Metodología

Para trabajar este proyecto, se utilizará una especificación de requerimientos en formato IEEE-830, así como datos de los estudiantes activos de la carrera integrados a una base de datos. Una vez realizado lo anterior se escogerá el marco de trabajo para lograr los objetivos antes mencionados, para este caso será SCRUM, debido a la forma iterativa de trabajar a través de ciclos conocidos como Sprints.

La duración de estos puede ir de 2 semanas hasta un mes, dependiendo de la complejidad de las tareas a realizar dando como resultado un producto listo para entregar. Dentro de cada sprint se contemplan actividades con respecto al análisis, diseño, desarrollo, prueba e integración al producto. Todo desarrollo SCRUM maneja lo siguiente:

- Retroalimentación o feedback: con cada sprint realizado, los clientes y el equipo de desarrollo durante una reunión, revisan el producto y se discuten cambios o futuras funcionalidades.

- Adaptación al cambio: los cambios se pueden realizar en cualquier punto del sprint para reforzar el valor del producto de acuerdo con la retroalimentación de los clientes.
- Satisfacción y aseguramiento de la calidad: al trabajar de forma colaborativa con el cliente se puede tener un producto con las características requeridas, satisfaciendo las necesidades para lo cual fue diseñado.
- Incremento funcional: al trabajar de forma iterativa y siendo el entregable de cada sprint un producto intermedio funcional, se puede aceptar, rechazar o replantear las funcionalidades de este.

1.4 Estado del arte

Al ser la deserción escolar una problemática que afecta a las instituciones de educación superior a nivel general, las investigaciones orientadas a esta han permitido observar desde una perspectiva computacional las distintas formas en las cuales, se ha tratado de dar con una solución que permita bajar la tasa de deserción en las instituciones.

La minería de datos ha tenido presencia en diferentes estudios con la finalidad de encontrar causas de la deserción escolar. En el caso particular de las Instituciones de Educación Superior públicas de México son distintas las investigaciones y muy variados los resultados alcanzados [3]. En la investigación realizada en la Universidad Tecnológica de Izúcar de Matamoros (UTIM) donde en el documento Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo de los k vecinos más cercanos [4] exponen los resultados obtenidos, que concluyen en una herramienta que permite calcular la probabilidad de deserción de cada uno de los estudiantes de la UTIM, apoyando de esta manera el proceso de tutorías de dicha institución.

En el Tecnológico de Estudios Superiores de Jocotitlán se realizó una investigación presentada en el documento Minería de datos aplicada para la identificación de factores de riesgo en alumnos [5], cuyo objetivo fue implementar un sistema que realiza más eficiente el proceso de tutorías. Para esto los investigadores analizaron información de 831 estudiantes de las generaciones de 2008 a 2013 de la carrera de Ingeniería en Sistemas Computacionales.

En la primera se usaron los datos de estudiantes inscritos y los que se dieron de baja donde los sometieron al proceso de selección, limpieza y transformación de datos para posteriormente emplear arboles de decisión. En el segundo caso emplearon técnicas de clustering y reglas de asociación, principalmente en los algoritmos K-means y A priori, ambas enfocadas al uso de la metodología KDD.

En [6] se implementó un proceso predictor de deserción aplicando minería de datos basado en redes neuronales, árboles de decisión y clustering, utilizando

como fuente de datos los registros históricos de su base académica, socioeconómica y psicopedagógica del periodo 2008-2013. En este caso se aplicó la metodología CRISP-DM (Cross Industry Standard Process for Data Mining) que consiste en un proceso con seis fases no necesariamente sucesivas que garantiza una mejor planeación y como consecuencia, mejores resultados. Una vez hecho este proceso, se procedió a la toma de decisiones con inteligencia de negocios para determinar una planeación dentro de las áreas administrativa, docente y psicopedagógica de la institución.

La investigación llevada a cabo en la Universidad del Zulia en Venezuela, hace énfasis en las etapas que realizó para su estudio sobre la deserción escolar aplicada a la Licenciatura en Computación perteneciente al Programa de Ciencia y Tecnología del Núcleo LUZ-NPF (Núcleo Punto Fijo), en la cual se nos menciona que la necesidad de obtener datos para identificar las causas de deserción de los estudiantes en los primeros semestres de su carrera, es indispensable para tomar las acciones pertinentes y poder disminuir este índice, y no menos importante, predecir su deserción en cualquier momento, a efectos de su monitoreo y así poder tomar de acciones correctivas [7].

Dicha investigación hace uso de la minería de datos haciendo uso de técnicas supervisadas y no supervisadas para la predicción de las causas de la deserción en la Licenciatura en Computación, tomando en cuenta a la metodología CRISP-DM, así como una muestra de 44 alumnos, integración de distintas bases de datos y algoritmos de árboles de decisión y k-vecinos para proseguir a utilizar el conocimiento obtenido en este proceso con Inteligencia de Negocios.

1.5 Resultados esperados

El resultado final será un sistema web que permita consultar información relacionada con los estudiantes de ingeniería en tecnologías de la información e identificar patrones para planear estrategias contra la deserción escolar.

Capitulo II. Marco Teórico

2.1 SCRUM

Scrum es un marco de trabajo ágil que permite trabajar proyectos de software de manera más eficiente, concentrándose en el software que, en la documentación, como se hacía en otras metodologías. En la siguiente figura se muestra el proceso Scrum.

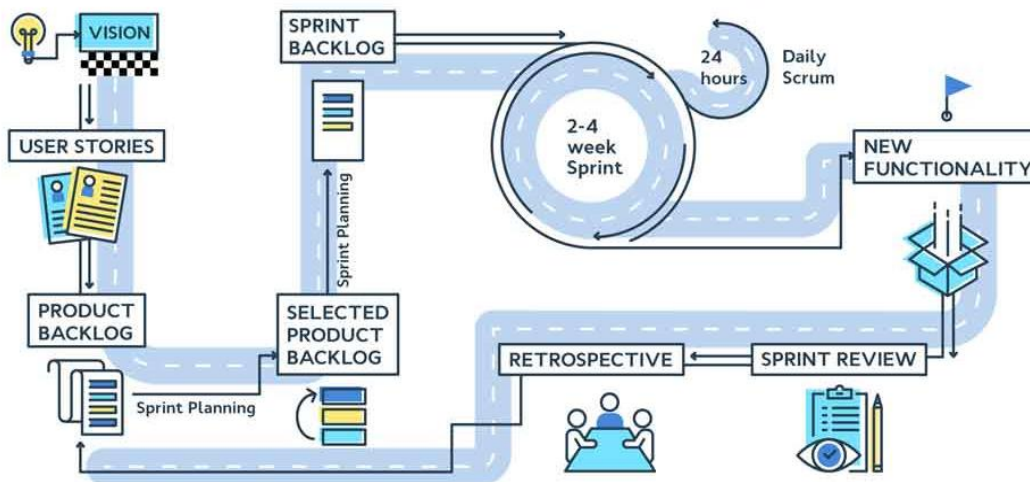


Figura 1. Marco de trabajo SCRUM.

2.1.1 Equipo Scrum

El equipo Scrum debe ser autogestionado y multifuncional, diseñado para optimizar la flexibilidad, la creatividad y la productividad, además de contemplar 3 roles existentes en este proceso, que se describen a continuación.

2.1.2 Product Owner

También conocido como dueño del producto, representa la voz del cliente, y es el encargado de maximizar el valor del producto. Es responsable de la viabilidad de este y del producto backlog, que contiene generalmente la visión general del proyecto en pequeñas tareas.

2.1.3 Scrum Master

Aquella persona que se encarga de liderar al equipo, guiándolo para que cumpla las reglas y los procesos de la metodología. Es responsable de eliminar los impedimentos, facilitar los procesos, gestionar y asegurar que el proceso de desarrollo se lleve a cabo de forma correcta.

2.1.4 Development Team

Son todas y cada una de las personas requeridas para la construcción del producto en cuestión, las características de este equipo es que, debe ser auto gestionado y auto organizado.

2.2 MVC

La arquitectura Modelo Vista Controlador se centra en el desarrollo de aplicaciones complejas a través de la independencia de los componentes como lo son el modelo de datos, la vista y el controlador que es el encargado de comunicar a los dos anteriores.

2.3 PHP

En la página oficial de PHP nos define que PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

2.4 SQL

Las siglas SQL corresponden a Structured Query Language, es un lenguaje estándar que permite manejar los datos de una base de datos relacional (Marqués, 2011).

2.5 CodeIgniter

Según la página oficial de CodeIgniter, es un potente marco PHP con una huella muy pequeña, creado para desarrolladores que necesitan un conjunto de herramientas simple y elegante para crear aplicaciones web con todas las funciones.

2.6 Grocery CRUD

Se nos informa en el sitio oficial de Grocery CRUD que es una biblioteca de código abierto que facilita la vida de un desarrollador. Con pocas líneas simples de código, puede crear un CRUD completamente estable con buenas vistas. ¡Un sistema completamente automático con el que incluso un novato en PHP puede trabajar!

2.7 Balsamiq Wireframes

Según la página oficial de Balsamiq, es una herramienta rápida de estructura de líneas de interfaz de usuario de baja fidelidad que reproduce la experiencia de dibujar en un bloc de notas o pizarra, pero usando una computadora.

2.8 Figma

Es un software para el maquetado de alta fidelidad basado en la web la cual trabaja con gráficos vectoriales y creación de prototipos que se pueden trabajar colaborativamente.

2.9 Enterprise Architect

Desarrollado por Sparx Systems, Enterprise Architect es una herramienta gráfica multiusuario diseñada para ayudar a construir sistemas robustos y fáciles de mantener. Admite UML, BPMN, Bases de Datos, Ingeniería Inversa, entre otros.

2.10 Chart.js

Es una biblioteca de código abierto que permite dibujar graficas dinámicas por medio de JavaScript, es acoplable con tecnologías frontend como Angular, Vue JS o React.

2.11 Node.js

Dentro de la página oficial de Node.js nos explican que fue Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables, se comporta de una forma similar a JavaScript en el navegador - el bucle de eventos está oculto al usuario y es propiedad de Open JS Foundation.

2.12 Jest

En el sitio oficial de este framework se nos dice que Jest es un marco de prueba de JavaScript encantador que se centra en la simplicidad. ¡Funciona con proyectos que usan: Babel, TypeScript, Node, React, Angular, Vue y más!

Capitulo III. Análisis y Diseño

3.1 Especificación de requerimientos

En esta sección se explicará con detalle los requerimientos funcionales y no funcionales que serán bases para realizar el análisis del sistema. Los requerimientos que aquí se especifican deben cumplir en la implementación de este. A continuación, se presenta un formato IEE-830 que será la base de esta especificación.

Especificación de Requisitos según el estándar de IEEE 830

1. Introducción

Una problemática común dentro de las instituciones de educación superior ha sido la deserción escolar, la cual se puede presentar por la presencia de factores de distinta índole.

Existen varios estudios realizados, con la finalidad de identificar las causas de la deserción escolar; sin embargo, los orígenes son multifactoriales y, por lo tanto, las propuestas de solución tienen diferentes resultados. Con base en la Guía para la Autoevaluación de Programas de Educación Superior (GAPES) de los Comités Interinstitucionales para la Evaluación de la Educación Superior (CIEES), la carrera de Tecnologías de la Información la eficiencia terminal (ET), para una cohorte 2012, 2013 fue de 29.23% y 22.82% respectivamente. Los datos de la Asociación Nacional de Universidades e Instituciones de Educación

Superior (ANUIES) reporta que en el país la eficiencia terminal en educación superior del ciclo 2016-2017, oscila entre el 30 y 40%. Para el caso de las universidades y tecnológicos que ofertan la carrera en tecnologías de la Información, la eficiencia terminal está en un 24%. La Ingeniería en Tecnologías de la Información de la BUAP tiene en promedio de 24.86% de ET para sus primeras dos cohortes, pero se pondera como deseable que éste se incremente al 30% con el fortalecimiento del acompañamiento de los estudiantes mediante tutorías y la atención a la deserción escolar. Sin embargo, la crisis sanitaria y económica que sufre el país ocasionado por el coronavirus COVID-19, ha propiciado reuniones entre los líderes de economía y finanzas para buscar una recuperación económica en los países como el nuestro; en consecuencia, se agregan nuevos factores de aumento en los índices de deserción escolar propiciados por la brecha digital que impacta en la educación. La UNESCO, ha señalado de manera acertada que el cierre de los centros educativos en todo el país tendrá consecuencias muy negativas en los ámbitos sociales y económicos. Debido, a la interrupción del aprendizaje, la desigualdad en las plataformas de aprendizaje digital, el incremento en los costos de los servicios de internet y adquisición de equipos electrónicos.

A continuación, se proporcionará una introducción a todo el documento de Especificación de Requisitos del Software (ERS).

1.1 Propósito

El propósito principal de este documento es definir y explicar los requerimientos funcionales y no funcionales, usuarios y restricciones para el sistema que aquí se especifica, dirigido a los stakeholders interesados.

1.2 Ámbito del sistema

El sistema que aquí se define se llamará **Sistema Estadístico para identificar causas de deserción escolar (SEICDE)**.

El sistema realizará lo siguiente:

- Mostrar los cortes generacionales existentes, desde la primera generación registrada.
- Leer los historiales académicos de cada estudiante que se encuentre activo, en suspensión temporal o haya desertado, así como los ya egresados para almacenarlos en la base de datos.
- Mostrar un marcado de materias en los “mapas gráficos” de los planes Minerva cuatrimestral (2012 - 2015) y semestral (2016 - actual) para cada estudiante donde se marquen materias aprobadas, pendientes y no cursadas.
- Crear y mostrar reportes por materia en base a la comparación entre dos periodos escolares existentes.

- Crear y mostrar reportes por corte generacional para ver el comportamiento de la base estudiantil que pertenezca a esta.
- Exportar en Excel las gráficas y datos que se muestren en los reportes.
- Ingresar con usuario y contraseña (que será el número de trabajador del coordinador(a) en turno).

El sistema no realizará lo siguiente:

- Eliminar registros directos de estudiantes de cualquier corte generacional.
- Enviar reportes a otra persona que no sea el coordinador de carrera.
- Modificar manualmente datos sensibles de los estudiantes como lo son nombre completo, matrícula o alguna calificación de materias aprobadas.
- Mostrar información sobre los estudiantes de forma externa, solo al coordinador.

Los beneficios que tendrá son:

- Visualizar la información académica del estudiante de una forma más gráfica y entendible, puesto que se obtienen directo del cardex de este y se muestran en el mapa gráfico que proporciona la FCC.
- Realizar reportes de materias por periodo y cortes generacionales para observación, análisis y comportamiento de los promedios generales.

Los objetivos y metas por alcanzar son:

- Proporcionar un base para buscar patrones en los datos conforme a los promedios a nivel materia (por comparación de periodos existentes) y a nivel generación (promedios generales del estudiante correspondiente al corte).
- Elaborar un dashboard útil y práctico para el manejo y tratamiento de la información.
- Estructurar reportes entendibles para usar la información recabada en una toma de decisiones en una etapa temprana.

1.3 Definiciones, Acrónimos y Abreviaturas

SEICDE: Acrónimo de Sistema Estadístico para identificar causas de deserción escolar.

BUAP: Benemérita Universidad Autónoma de Puebla.

FCC: Abreviatura de Facultad de Ciencias de la Computación.

ITI: Abreviatura de Ingeniería en Tecnologías de la Información.

Dashboard: Panel de control de información.

Cardex: Nombre que se le da al historial académico de un estudiante.

BANNER: Sistema institucional que maneja los procesos dentro de la institución.

DOM: Abreviatura de Document Object Model.

1.4 Referencias

Especificación de Requisitos según el estándar de IEEE 830 IEEE Std. 830-1998.

1.5 Visión General del Documento

A continuación, en las secciones posteriores se va a manejar al sistema como un producto de software, así como los factores que influyen en su elaboración, contexto en el que va a existir, entre otros.

2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto.

2.1 Perspectiva del Producto

SEICDE será un sistema independiente de uso exclusivo para la coordinación de la carrera de ITI dentro de la FCC-BUAP, no se vinculará al sistema BANNER de la BUAP.

2.2 Funciones del Producto

Dentro de las funciones se destaca:

- El sistema creará distintos reportes (esta función se encargará de crear reportes a partir de los datos obtenidos en el cardex que incluirán gráficas lineales, tablas con datos y exportación a Excel).
- El sistema permitirá cargar el cardex del estudiante (esta función se encargará de extraer los datos de los cardex que se encuentren en formato PDF o HTML).
- El sistema visualizará los mapas gráficos del plan cuatrimestral y semestral de la carrera de ITI (esta función se encargará de marcar las materias aprobadas, pendientes y cerradas de cada estudiante en base a, los datos de su cardex para ver su avance académico).

2.3 Características de los Usuarios

Se describen el o los usuarios que van a interactuar con el sistema.

Usuario	Descripción general
<i>Coordinador de carrera</i>	El nivel educacional de este usuario no es definido, debido a que puede variar (pueden tener maestría o doctorado),

debe tener conocimientos de bases de datos y en el manejo de los procesos dentro de la coordinación.
Cuenta con alto nivel de acceso a la información y puede manipular el sistema según lo necesite.

2.4 Restricciones

Las restricciones que se contemplan son las siguientes:

- Ninguna persona que no sea el coordinador(a) en turno (o en su defecto persona que autorice) no pueden acceder al sistema.
- Los cardex deben estar en formato PDF o HTML para su tratamiento, extracción y carga de datos.
- El framework para desarrollar este sistema será CodeIgniter en su versión 4.
- La base de datos se manejará en MySQL.
- El manejo de las claves de cada materia debe ser por separado, puesto que hay materias que no son aplicables al plan Minerva semestral (que entró en vigor en la generación 2016), cambiaron su cantidad de créditos o área a la que pertenecían en el Minerva cuatrimestral.
- Al ser de uso institucional, debe manejar los colores de la dependencia.
- Al tratarse de un sistema web, debe contar con un dominio de uso privado.
- El repositorio donde se encuentre el proyecto debe ser privado y solo podrá tener acceso el coordinador en turno y el equipo de desarrollo.
- Desarrollo bajo el modelo MVC.
- Uso del framework CodeIgniter 4.0
- Lenguajes de programación: PHP (por parte del framework) y JavaScript (para comunicación y manejo a nivel de DOM).

2.5 Suposiciones y Dependencias

Dentro de estas se debe considerar:

- A nivel de navegador, si se usa Chrome o Firefox, que en su versión 100 se especifican nuevos cambios al despliegue de sistemas y/o páginas web que pueden afectar el despliegue de este sistema.
- En caso de existir un nuevo(a) coordinador(a), hacer el traspaso correspondiente de este sistema.
- Las materias de tipo DESIT deben revisarse antes de cargarse a la base de datos.
- Si existe un cambio en los planes de estudio, revisar los ajustes a nivel tabla de la base de datos y sus relaciones.

2.6 Requisitos Futuros

Dentro de estos se contempla una vinculación con las otras carreras que se ofertan dentro de la FCC que son Ingeniería en Ciencias de la Computación y Licenciatura en Ciencias de la Computación, con el fin de obtener una población más grande para poder depurar y obtener datos de mejor calidad para el análisis.

3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir diseñar un sistema que satisfaga estos, todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas.

3.1 Interfaces Externas

Para la realización e implementación del software de tal forma que sea eficiente y con la menor cantidad de fallas posible, se recomienda tomar en cuenta los siguientes aspectos.

Hardware:

En esta parte se contempla que se necesitan los equipos de cómputo estén en perfecto estado, con las siguientes características:

- Teclado.
- Mouse.
- Procesador de 1.60 GHz o superior.
- Memoria RAM de 8 GB en adelante.
- Conexión Wi-Fi y/o cable de red.
- Puertos USB 3.0
- Disco Duro de 1TB o superior.

Software:

- Navegador Web (de preferencia Chrome, Firefox o Safari).
- Sistema Operativo Windows 10/ macOS/ Linux.
- Servidor para desplegar el sistema (de preferencia institucional).

3.2 Funciones

El sistema deberá tener las siguientes funciones para uso exclusivo del coordinador ITI en turno:

<i>Usuario</i>	<i>Funciones Relacionadas</i>
<i>Coordinador ITI</i>	Iniciar Sesión: El coordinador debe iniciar sesión para poder acceder al sistema por medio de su número de trabajador y su contraseña. El sistema deberá:

- Buscar el número de trabajador y contraseña del coordinador.
- Dar acceso a los datos de los estudiantes de forma interna.

Consultar corte generacional: El coordinador puede ingresar a una sección llamada “Estudiantes”, donde debe estar cargados los cortes generacionales y dentro de estos, los alumnos que pertenecen a cada generación.

El sistema deberá:

- Cargar todas las generaciones existentes que al menos tengan un alumno registrado.
- Mostrar alumnos dentro de la generación que se consulte.

Consultar historial académico: El coordinador puede buscar a un estudiante en específico dentro de la generación que se encuentre ubicado, para poder acceder a su información general y a su mapa gráfico de materias.

El sistema deberá:

- Mostrar la matricula, nombre, porcentaje, materias aprobadas, estatus y el mapa gráfico de materias del estudiante a buscar.
- Cargar en el mapa grafico las materias aprobadas, pendientes y en caso de existir, las que ya no puede cursar.
- Crear la materia dentro del mapa en caso de no estar en la precarga del cardex.
- Buscar al estudiante por nombre o matricula.

Registrar nuevo historial académico: El coordinador puede registrar a estudiantes nuevos dentro de las generaciones ya existentes o para crear una nueva.

El sistema deberá:

- Filtrar la matricula, nombre, carrera, periodo de inicio, número de materias aprobadas del estudiante a un formato legible para la base de datos.
- Registrar el NRC, sección, clave, calificación y recursos (en caso de existir) de cada materia junto con su periodo de aprobación.
- Registrar los datos del estudiante en la base de datos.

- Registrar materias no existentes (que no vienen en el currículo por default de la carrera, como lo son las DESIT) junto con sus datos en la base.
- Leer la carpeta donde se guardan los historiales académicos.

Generar reporte por materia:

El sistema deberá:

- Crear gráficas de los periodos de comparación.
- Crear tablas por periodo de comparación.
- Exportar a Excel el reporte generado.

Generar reporte por corte generacional:

El sistema deberá:

- Crear tablas por periodo de comparación.
- Exportar a Excel el reporte generado.

Extraer información de historial académico (archivo PDF o HTML):

El sistema deberá:

- Leer el archivo HTML o PDF.
- Transforma los datos a un formato legible para la base de datos.
- Extraer los “tokens” del archivo.
- Ejecutar el script de extracción.

3.3 Requisitos de Rendimiento

Dentro del rendimiento se toma en cuenta lo siguiente:

- Cada cardex cuenta con un numero **N** de materias cursadas, el rango máximo de materias cursadas por estudiante oscila entre las 52 a 56 materias aprobadas. Por lo tanto, el volumen de datos que se espera almacenar es mayor a los 1000 registros.
- Las consultas a la base deben ser óptimas, por el gran volumen que se va a manejar.
- La carga del marcado de materias puede tener un delay no mayor a 5 segundos.

3.4 Restricciones de Diseño

A nivel de interfaz, por tratarse de un sistema institucional, se piden manejar los colores en tonos azules y blancos, así como el inicio de sesión habilitado de forma exclusiva al coordinador(a) en turno.

3.5 Atributos del Sistema

El sistema deberá contar con:

- Usuario y contraseña para acceso, este pertenecerá al coordinador en turno.
- Permisos para acceso a los cardex almacenados a nivel de carpeta.
- Acceso controlado al repositorio del sistema en GitHub.

Todo lo anterior es referente a seguridad en una primera instancia. Al ser un sistema web, se infiere a que es un sistema portable que se puede consultar en un dominio establecido.

3.6 Otros Requisitos

No existen requisitos extras en el momento de la redacción de esta ERS.

4. Apéndices

Esta sección no es requerida al momento de la redacción de esta ERS.

3.2 Análisis del sistema

En esta sección específica se presenta el análisis del sistema desde un punto de vista estático, basado en los requerimientos y apoyado en el modelado UML (Unified Model Language) por medio de casos de uso y clases.

3.2.1 Casos de Uso

Actores

Al ser un sistema de uso privado (por la naturaleza de los datos), el actor principal será el coordinador de la carrera de ITI.

Diagrama de casos de uso

Dentro del análisis se puede encontrar el diagrama de casos de uso que nos permite ver de una forma más abstracta la interacción del usuario principal con el sistema en forma de acciones que este debe llevar a cabo. A continuación, en la figura 2 se propone el siguiente diagrama para este caso en específico:

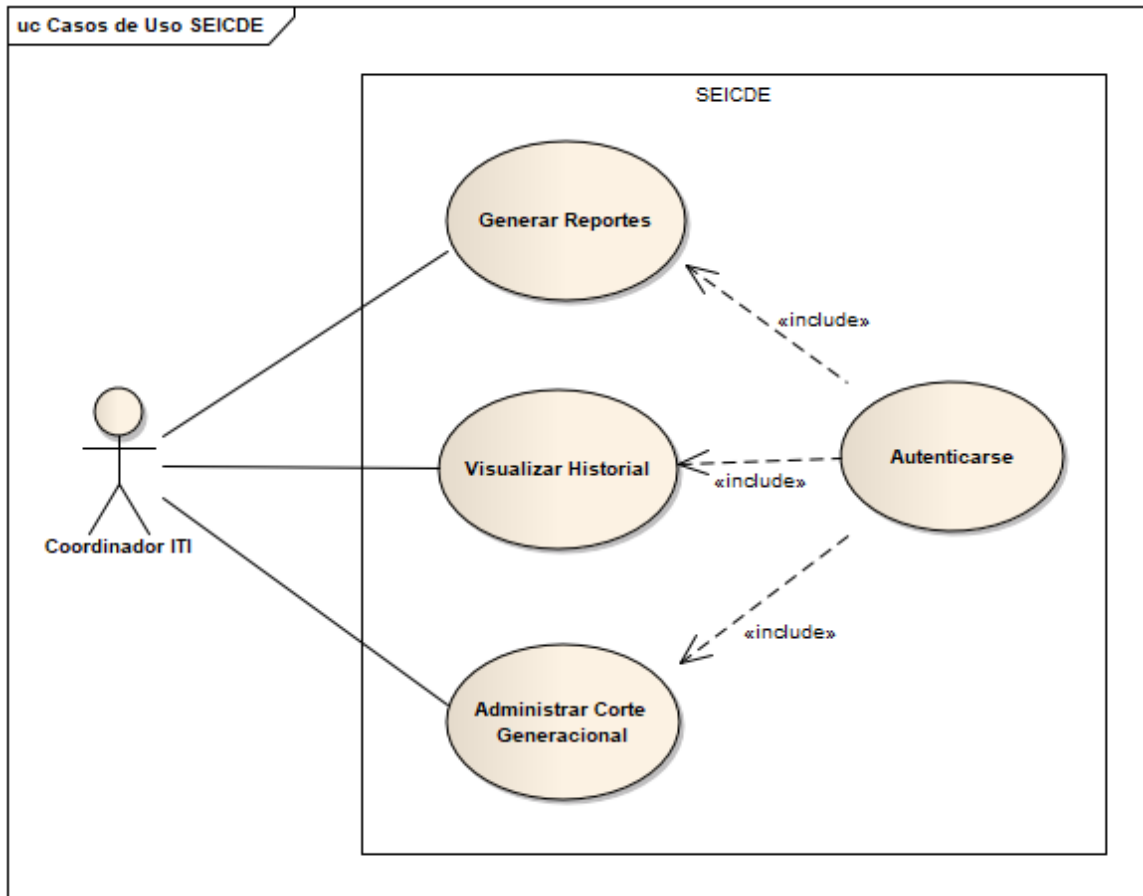


Figura 2. Diagrama de Casos de Uso

Especificación de casos de uso

Para cada caso de uso existe una descripción en la cual se conforma del caso nominal y en su mayoría de veces, un caso alternativo junto con los casos de error. A continuación, se describen los casos de uso mencionados en el diagrama anterior (Tabla 1 a 4).

Tabla 1. Descripción Caso de Uso Autenticarse.

Descripción del caso "Autenticarse"
<p>Identificación</p> <p>Nombre del caso de uso: Autenticarse</p> <p>Actor Principal: Coordinador ITI</p> <p>Actor(es) Secundario(s): Ninguno.</p> <p>Fecha de Elaboración: 08/11/2021.</p> <p>Fecha de Actualización: 11/11/2021.</p> <p>Responsable: Monica Portillo Sánchez.</p> <p>Versión: 1.0</p>
<p>Descripción nominal</p> <p>El coordinador entra al sistema con sus credenciales.</p>

Precondición:

El coordinador debe tener su número de trabajador.

1. El coordinador ingresa su número de trabajador y su contraseña.
2. El sistema valida los datos.
3. El sistema busca al usuario.
4. El sistema encuentra al usuario. E1
5. El sistema abre la ventana correspondiente.

Descripciones alternativas**Posibles errores****Error 1: No logra ingresar al sistema.**

El error comienza en el punto 4 de la descripción nominal

- 4.El sistema no encuentra al coordinador.
- 5.El coordinador vuelve a intentarlo después de 10 segundos.

Postcondiciones:

El sistema realiza lo siguiente:

- Permite el inicio de sesión a partir del número de trabajador del coordinador en turno.
- Carga las generaciones de ITI a partir del año 2012.

Descripción textual

El coordinador ingresa al sistema a través de su número de trabajador universitario que a su vez es su contraseña.

Tabla 2. Descripción Caso de Uso Visualizar Historial.

Descripción del caso “Visualizar Historial”**Identificación**

Nombre del caso de uso: Visualizar Historial.

Actor Principal: Coordinador ITI

Actor(es) Secundario(s): Ninguno.

Fecha de Elaboración: 08/11/2021.

Fecha de Actualización: 11/11/2021.

Responsable: Monica Portillo Sánchez.

Versión: 1.0

Descripción nominal

El coordinador quiere revisar un historial de un estudiante en específico.

Precondición:

El coordinador ya está dentro del sistema.

- 1.El coordinador entra en la sección “Estudiantes”.
- 2.El sistema muestra los cortes generacionales, (en los cuales existen al menos un registro).
- 3.El coordinador selecciona un corte generacional.
- 4.El sistema muestra la lista de estudiantes que pertenecen al corte.
- 5.El coordinador busca la matricula del estudiante.
- 6.El sistema muestra al estudiante que se buscó. E1
- 7.El coordinador selecciona al estudiante.
8. El sistema carga el mapa gráfico con las materias aprobadas y pendientes.

Descripciones alternativas

Posibles errores

Error 1: No muestra al estudiante.

El error comienza en el punto 6 de la descripción nominal

- 3.El sistema no muestra al estudiante que se buscó.
- 4.El coordinador intenta nuevamente con el nombre del estudiante.
- 5.El sistema marca la inexistencia del registro.

Postcondiciones:

El sistema realiza lo siguiente:

- Carga los cortes generacionales a partir del año 2012.
- Permite ver de forma visual las materias que ha cursado con éxito, las faltantes y las que no puede cursar el estudiante.
- Carga los estudiantes por corte generacional.

Descripción textual

El coordinador ingresa a la sección de estudiantes y escoge un corte generacional específico para poder facilitar la búsqueda y visualización del historial del estudiante a través del mapa gráfico que viene en el plan de estudios.

Tabla 3. Descripción Caso de Uso Generar Reportes.

Descripción del caso “Generar Reportes”

Identificación

Nombre del caso de uso: Generar Reportes.

Actor Principal: Coordinador ITI

Actor(es) Secundario(s): Ninguno.

Fecha de Elaboración: 08/11/2021.

Fecha de Actualización: 08/11/2021.

Responsable: Monica Portillo Sánchez.

Versión: 1.0

Descripción nominal

El coordinador quiere generar un reporte ya sea por corte generacional o por materia.

Precondición:

El sistema tiene cargados los periodos, las materias y los cortes generacionales existentes.

- 1.El coordinador ingresa a la sección de reportes.
- 2.El coordinador selecciona reporte por materia. A1
- 3.El sistema muestra las materias existentes y los periodos a comparar.
- 4.El coordinador selecciona la materia y los periodos a comparar.
- 5.El sistema carga la gráfica de promedios y la tabla con la información de la materia.
- 6.El sistema muestra los datos.

Descripciones alternativas

Alternativa 1: El coordinador quiere generar un reporte por corte generacional.

La alternativa empieza en el punto 2 de la descripción nominal.

- 2.El coordinador selecciona reporte por corte generacional.
 - 3.El sistema solicita que seleccione una generación.
 - 4.El coordinador selecciona la generación.
- Se regresa al punto 5 de la situación nominal.

Posibles errores

Error 1: No muestra los periodos.

El error comienza en el punto 3 de la descripción nominal

- 3.El sistema no muestra los periodos para comparar.
- 4.El coordinador cierra la ventana y vuelve a intentar después de 5 segundos.
- 5.El sistema no carga los periodos y muestra el mensaje de error.

Error 2: No muestra las materias.

El error comienza en el punto 3 de la descripción nominal

- 3.El sistema no muestra las materias existentes.
- 4.El coordinador cierra la ventana y vuelve a intentar después de 5 segundos.
- 5.El sistema muestra el mensaje de error en la carga de materias.

Postcondiciones:

El sistema realiza lo siguiente:

- Carga los periodos escolares existentes en la base de datos.
- Carga las materias existentes del plan semestral como del cuatrimestral.
- Dibuja una gráfica lineal que muestra los promedios de una materia en específico y de los periodos solicitados.
- Muestra una tabla con la información de la materia y sus promedios.

Descripción textual

El coordinador ingresa a la sección de reportes y puede realizar por corte generacional o por materia, teniendo en cuenta en este último caso, que debe elegir dos periodos a comparar, una vez hecho esto, se dibuja una gráfica lineal con los promedios apoyado en tablas que permiten ver los datos que se manejan.

Tabla 4. Descripción Caso de Uso Administrar Corte Generacional.

Descripción del caso “Administrar Corte Generacional”

Identificación

Nombre del caso de uso: Administrar Corte Generacional.

Actor Principal: Coordinador ITI

Actor(es) Secundario(s): Ninguno.

Fecha de Elaboración: 08/11/2021.

Fecha de Actualización: 08/11/2021.

Responsable: Monica Portillo Sánchez.

Versión: 1.0

Descripción nominal

El coordinador sube, modifica los historiales académicos, los cuales permiten identificar los cortes generacionales y los estudiantes que pertenecen al mismo.

Precondición:

El coordinador cuenta con los historiales en formato PDF o HTML.

1. El coordinador agrega los historiales académicos dentro de la carpeta hecha para este fin. A1
2. El sistema lee el archivo de la carpeta. E1
3. El sistema extrae los datos del archivo (nombre, matricula, materia aprobada, calificación, sección, NRC, periodo y en caso de existir, recurso de materia).
3. El sistema examina y transforma los datos del archivo.
4. El sistema registra los datos en la base de datos.

Descripciones alternativas

Alternativa 1: El coordinador tiene el historial en formato PDF.

La alternativa empieza en el punto 1 de la descripción nominal.

2. El sistema formatea el archivo a formato HTML.

Se regresa al punto 3 de la situación nominal.

Posibles errores

Error 1: El sistema no lee los historiales académicos.

El error comienza en el punto 2 de la descripción nominal

2. El sistema marca un error en el archivo a nivel de carpeta.

3. El sistema revisa el formato del archivo.

4. El sistema intenta leer nuevamente el archivo.

5. El sistema se detiene y pausa el registro de los demás archivos.

Postcondiciones:

El sistema realiza lo siguiente:

- Formatea los archivos a HTML para su análisis.
- Extrae datos del historial académico.
- Transforma y examina los datos a un formato aceptable para la base de datos.
- Realiza registros en la base de datos.

Descripción textual

El coordinador sube los historiales en una carpeta, la cual se lee a través de un script interno que manipula los historiales académicos para posteriormente extraer, transforma y cargar los datos que contienen en la base de datos.

3.2.2 Diagrama de clases

Uno de los diagramas más utilizados en análisis es el diagrama de clases. Nos permite ver como las entidades participantes interactúan entre sí y nos ofrece el panorama más cercano a lo que vería en una base de datos, solo con la diferencia en la abstracción de este diagrama. En la figura 3 se muestra el diagrama propuesto para este caso:

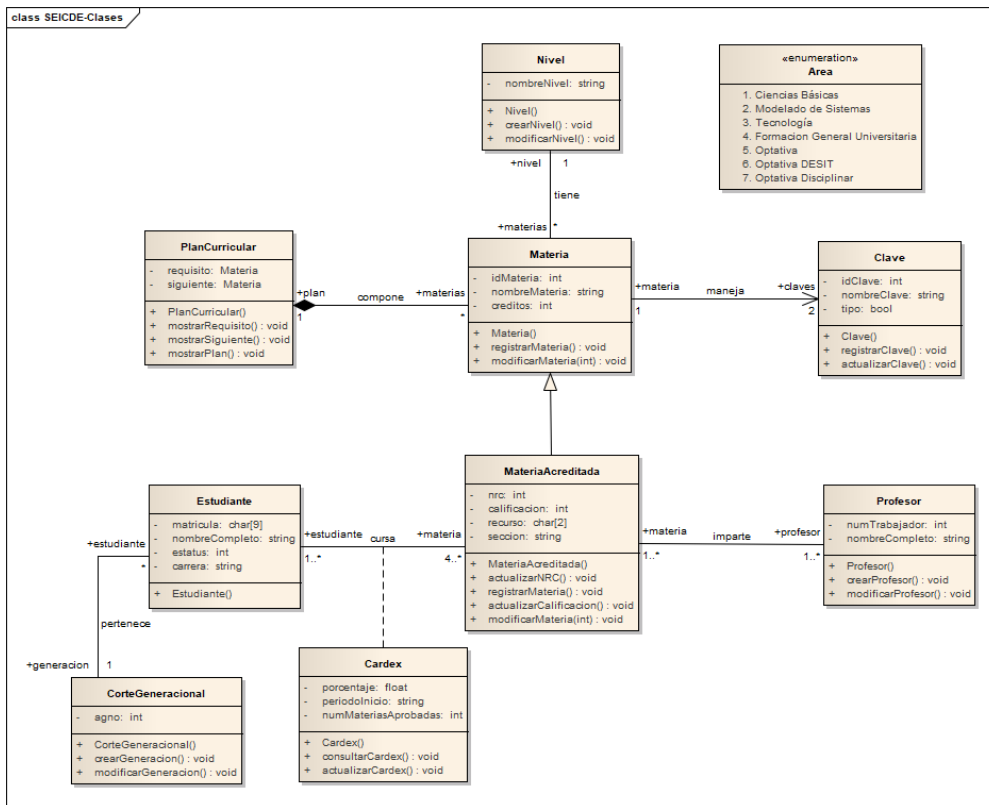


Figura 3. Diagrama de Clases

La justificación del modelo se explica a continuación:

Plan Curricular

Esta clase en sí misma es una composición de Materia, debido a que en permite el manejo de las materias de acuerdo con el plan que se requiera consultar (cuatrimestral o semestral) a través de una abstracción más sencilla como lo es el conocimiento del requisito y la materia siguiente.

Materia

Esta es una de las clases base, permite tanto la existencia del plan curricular, así como el manejo de sus respectivas claves. Sobre de esta se mueve una generalización de la parte en la que el estudiante cursa y aprueba la misma.

Clave

Esta clase permite la existencia de las claves de cada materia debido a que, al manejar dos planes curriculares diferentes uno de otro, las claves se representan de forma diferente también en la cantidad de créditos y se maneja un atributo booleano para diferenciarlas.

Estudiante

La clase permite relacionar al estudiante en turno dentro de la matrícula ITI con las materias que va acreditando a lo largo de la carrera, debido a que cada uno tiene calificaciones diferentes y permite conocer el cardex de este.

Materia Acreditada

A través de su generalización con Materia, esta clase hija permite manejar la parte de calificaciones, recursos y otros datos presentes en la aprobación de materias

Profesor

Esta clase se vincula con las materias acreditadas con fines informativos con respecto a quien fue quien impartió esa materia en cierto periodo escolar.

Cardex

También conocido como historial académico, sirve para asociar las materias acreditadas con el respectivo estudiante que las cursa, cargando únicamente con el periodo de inicio, porcentaje y cantidad de materias que lleva el estudiante al momento de consultar el cardex.

Área

Esta clase enumerada permite abstraer de forma eficaz las áreas con las se trabaja dentro de los planes curriculares, debido a que, muchas materias pertenecen a más de un área y es fácil establecer una vinculación entre estas.

Corte Generacional

Esta clase representa la generación a la que pertenece el estudiante, permite el manejo de los grupos masivos de estudiantes y al trabajar a profundidad con este, hace más fácil el acceso a información específica en conjunto con las otras clases.

3.2.3 Interfaces Gráficas

Baja fidelidad

Las interfaces en baja fidelidad son aquellas con una abstracción más general en las cuál solo se destacan aspectos básicos como botones, inputs, imágenes “vacías” y comportamientos básicos, que sirven como base para las de alta fidelidad, asimismo no toma en cuenta los colores y la tipografía de forma temporal. A continuación, se muestran las interfaces de baja fidelidad para este sistema:

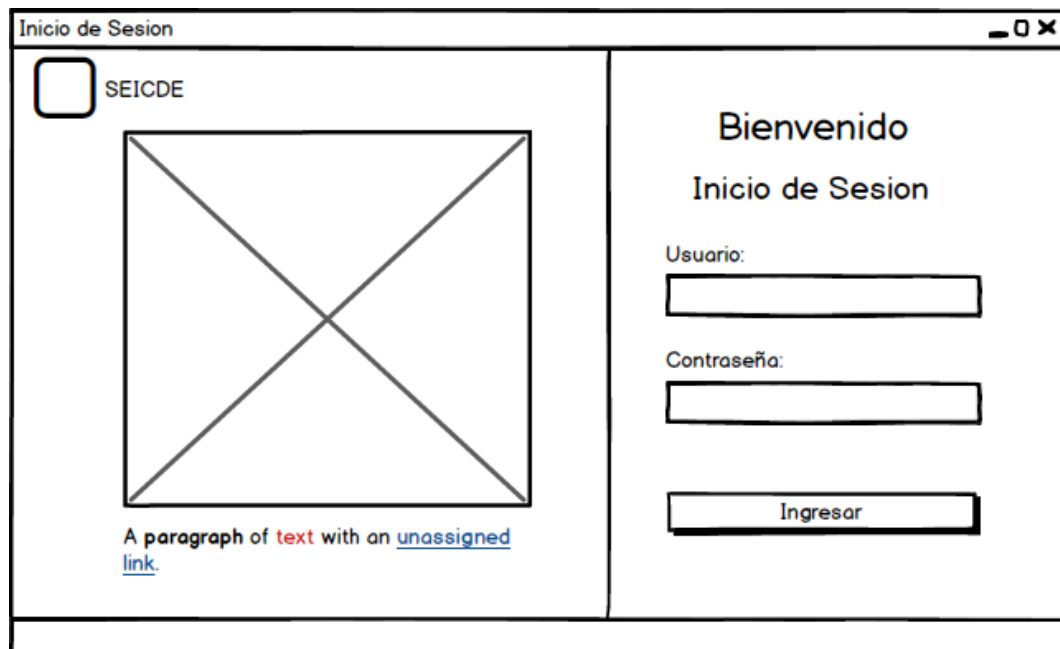


Figura 4. Prototipo Inicio de sesión.

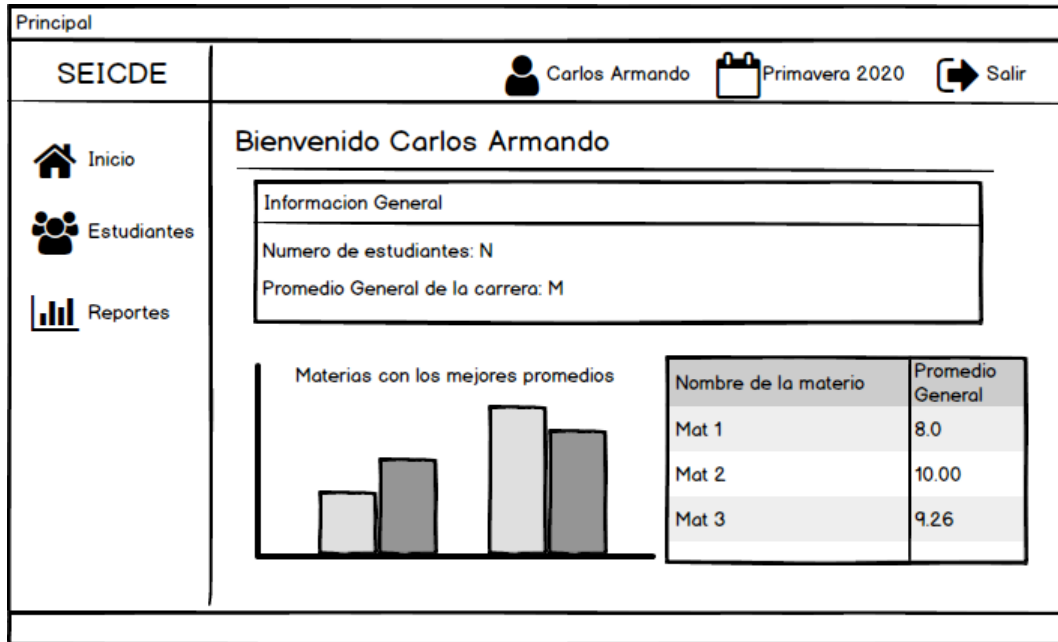


Figura 5. Prototipo Ventana Principal

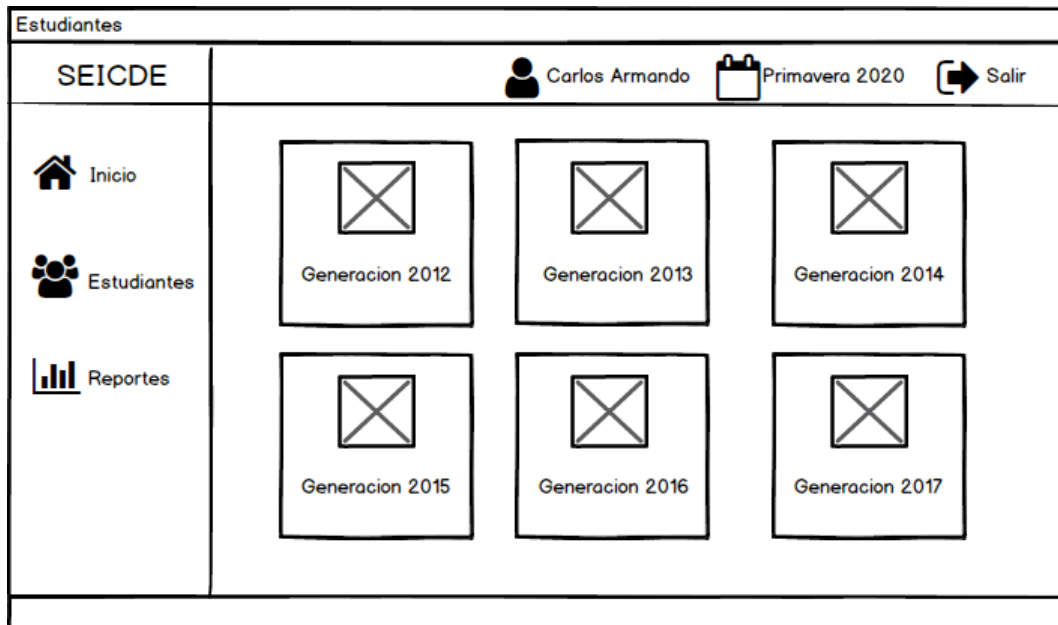


Figura 6. Prototipo de sección "Estudiantes"

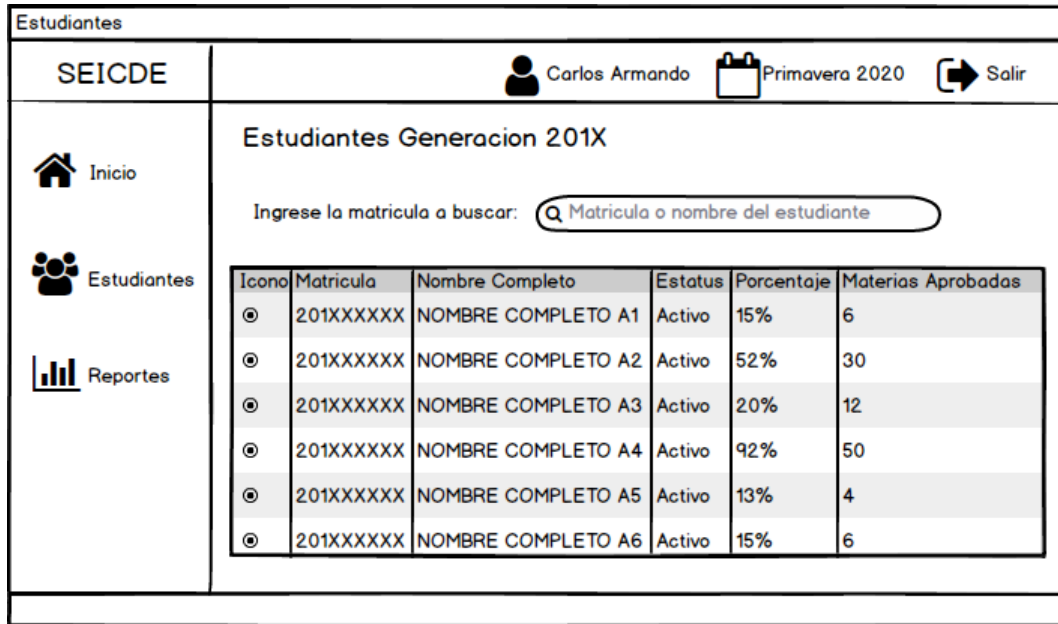


Figura 7. Prototipo de listado de estudiantes de X generación

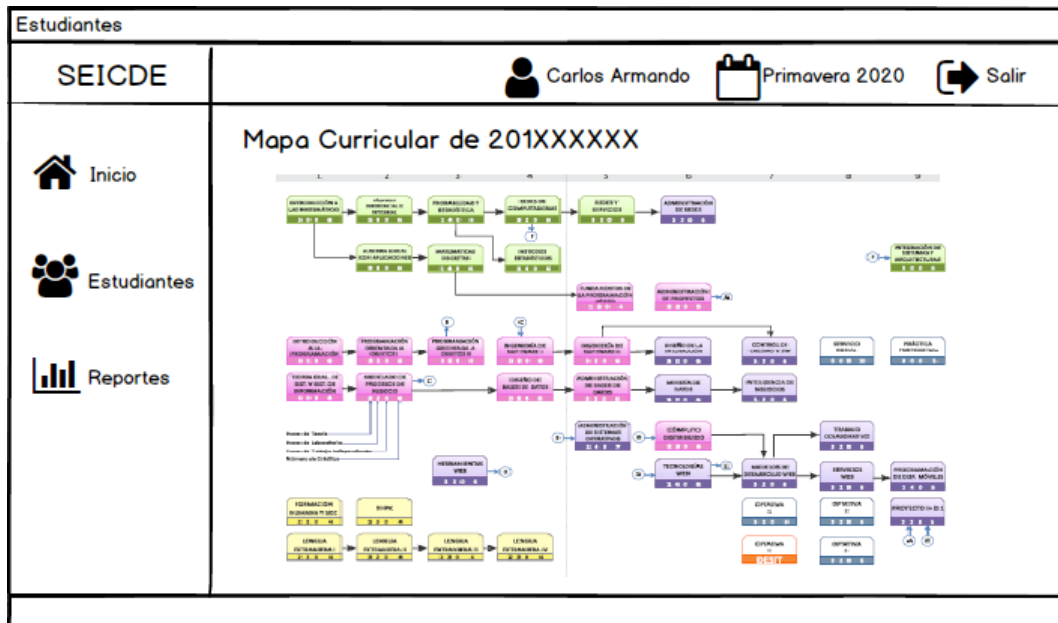


Figura 8. Prototipo de Mapa Curricular

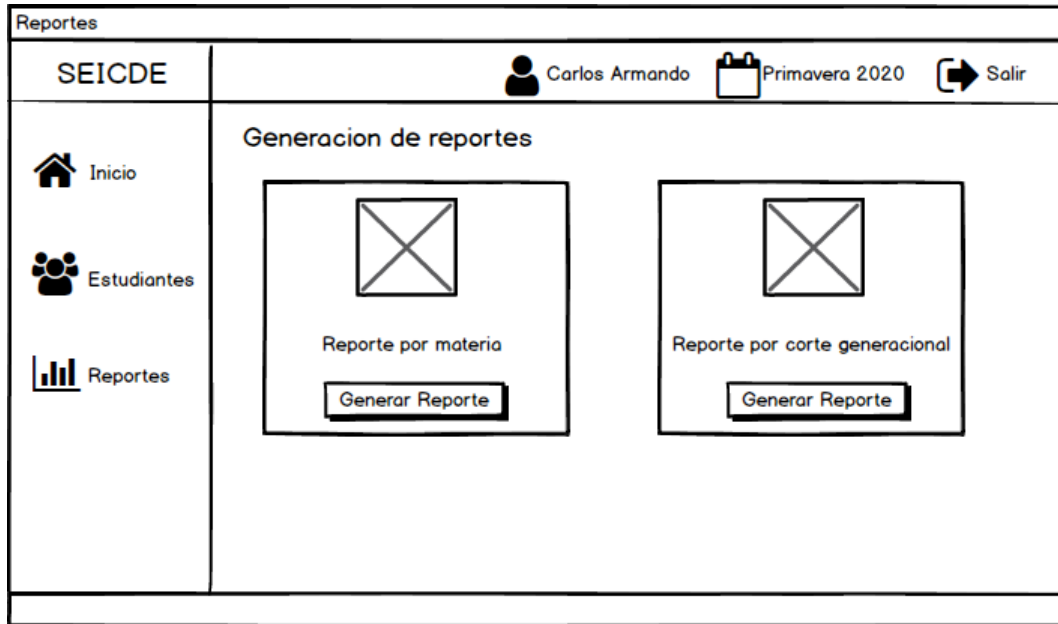


Figura 9. Prototipo de sección "Reportes"

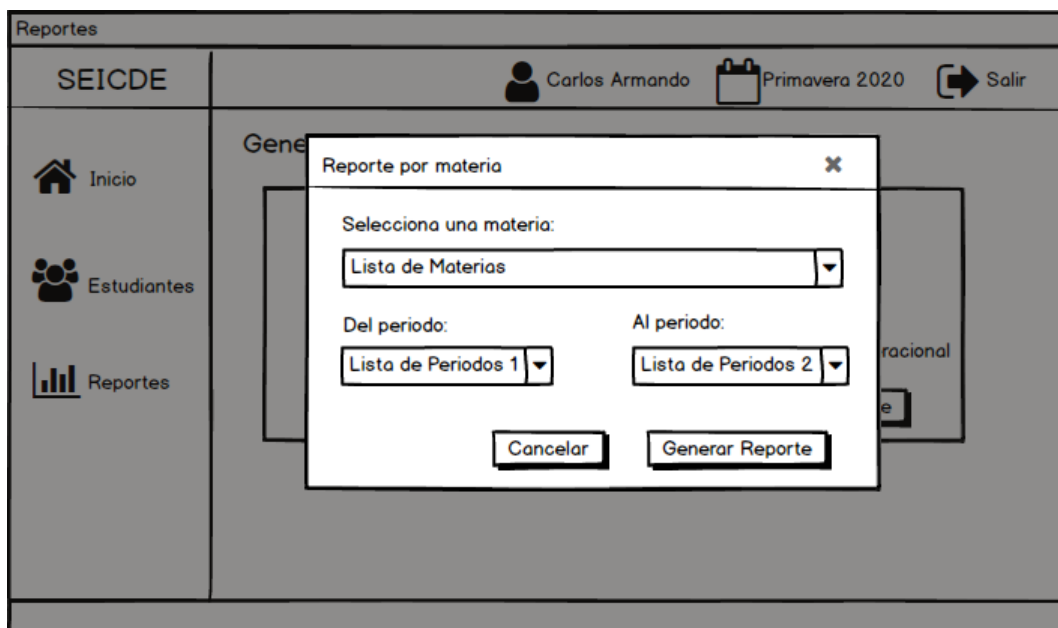


Figura 10. Prototipo de modal para Reporte por Materia

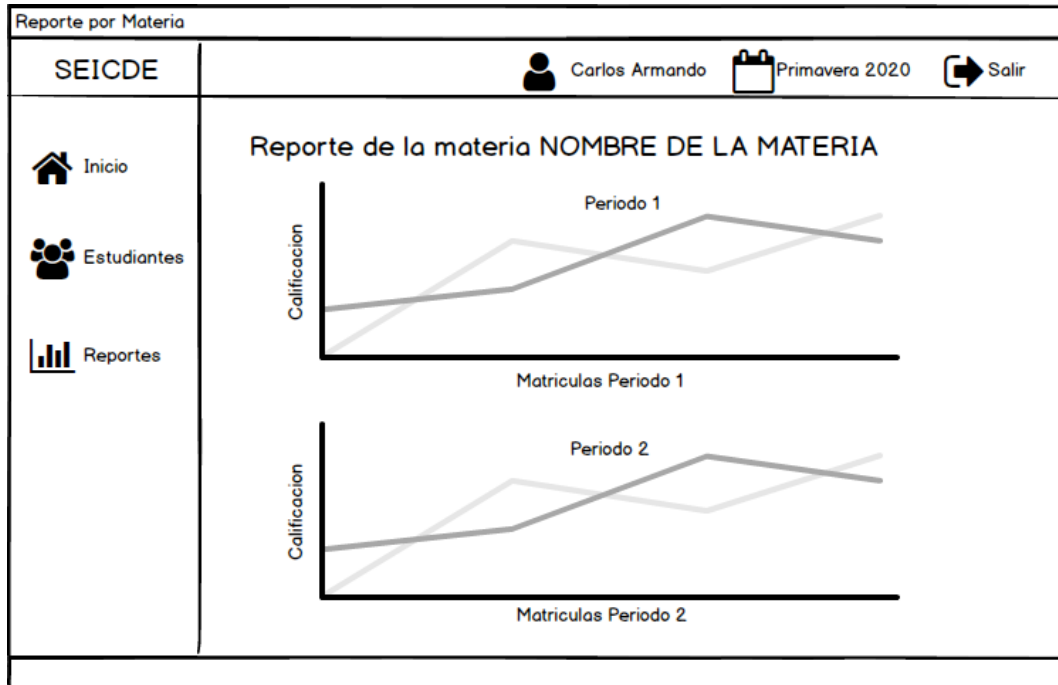


Figura 11. Prototipo de Reporte por materia - Sección de gráficas

Alta fidelidad

Las interfaces en alta fidelidad son aquellas que son más apegadas a como se verán en la codificación y despliegue del sistema. En las figuras se muestran las interfaces realizadas en Figma. Tómese en cuenta que, algunas imágenes solo sirven de referencia, cambian una vez en la implementación.



Sistema Estadístico para identificar causas de deserción escolar

Bienvenido
Inicio de Sesión

Usuario

XXXXXXXXXX

Contraseña

Ingresar

Figura 12. Interfaz de alta fidelidad - Inicio de Sesión

SEICDE
Carlos Armando
Primavera 2020
Salir

Inicio

Estudiantes

Reportes

Bienvenido Carlos Armando

Información General

Número de estudiantes: N

Promedio general de la carrera: M

Materias con promedios altos

#	Nombre de la materia	Promedio
1	Materia A	N
2	Materia B	N
3	Materia C	N
4	Materia D	N
5	Materia E	N

Figura 13. Interfaz de alta fidelidad - Principal

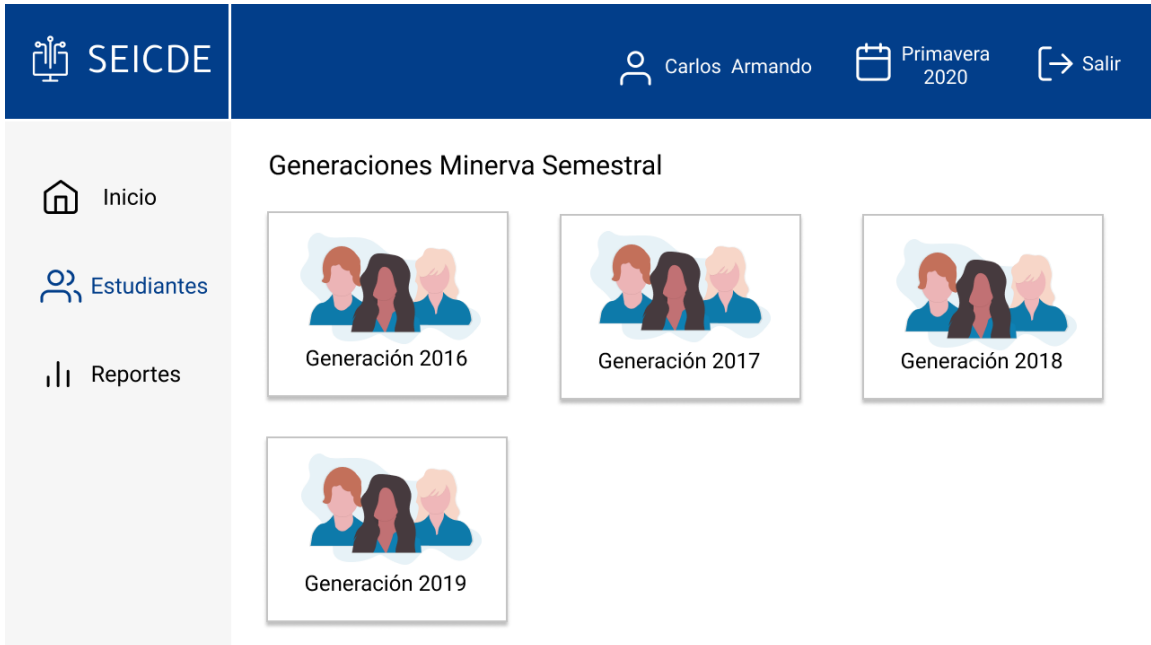


Figura 14. Interfaz de alta fidelidad - Estudiantes

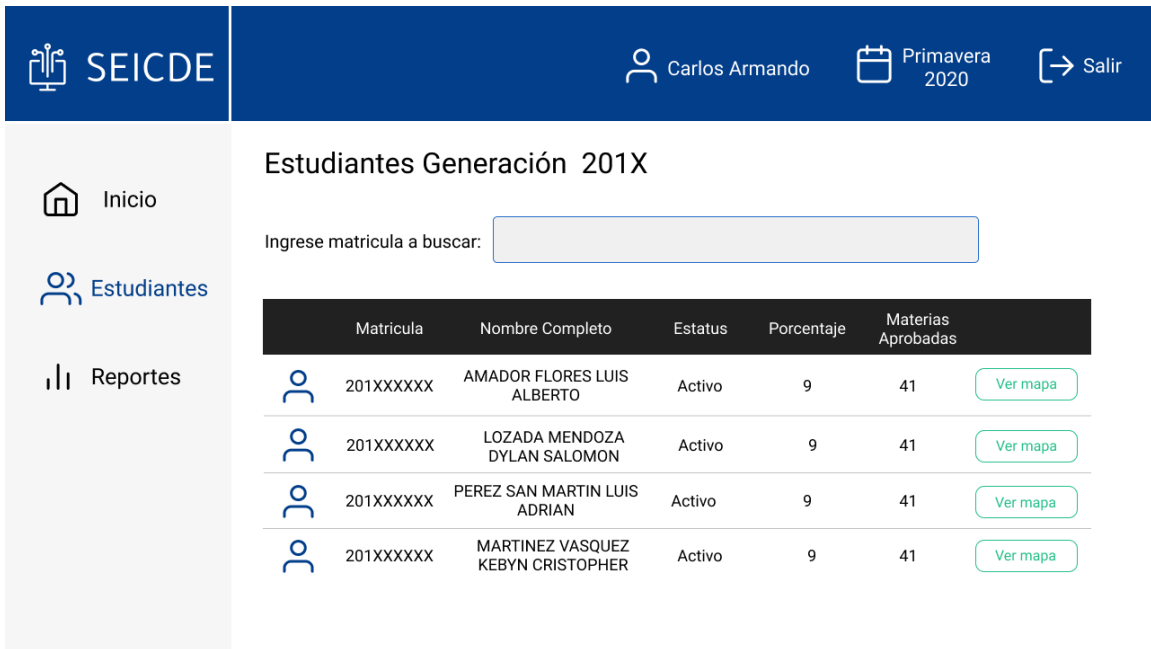


Figura 15. Interfaz de alta fidelidad - Lista de estudiantes

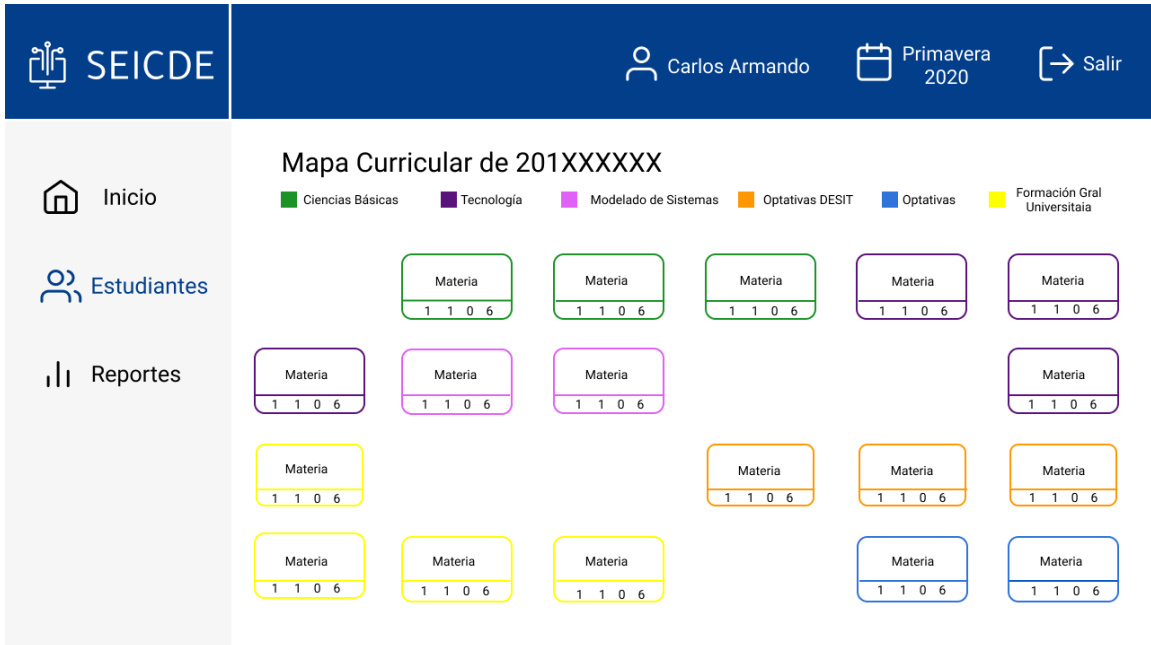


Figura 16. Interfaz de alta fidelidad - Mapa gráfico sin marcar

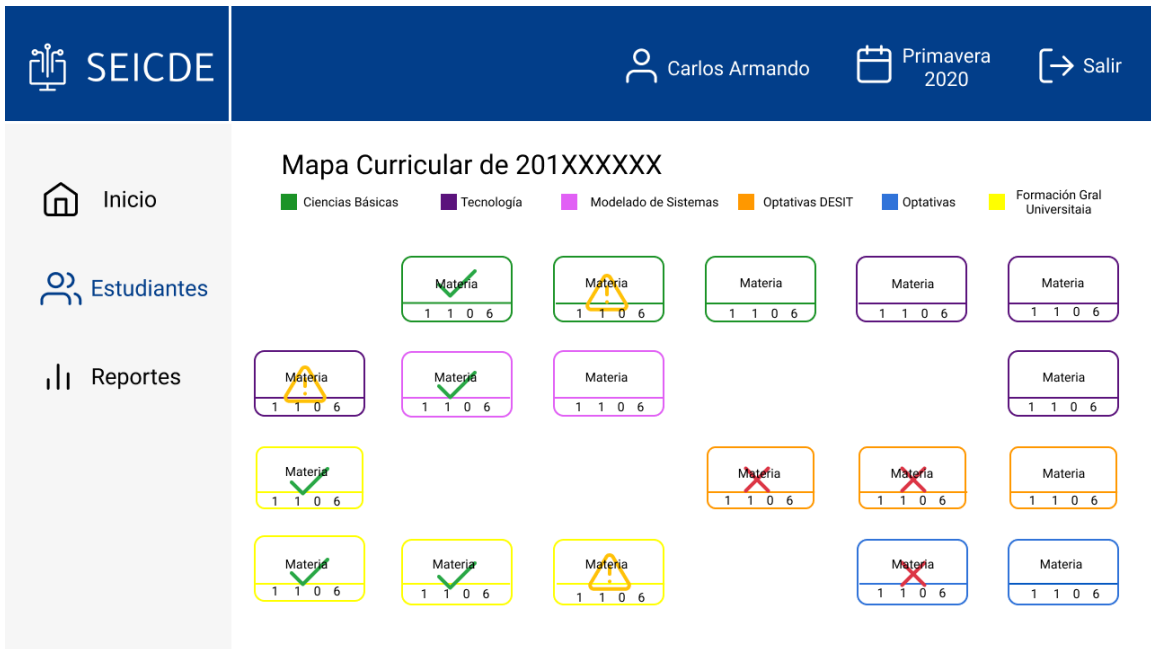


Figura 17. Interfaz de alta fidelidad - Mapa gráfico marcado



Figura 18. Interfaz de alta fidelidad - Menú Reportes

3.3 Desarrollo SCRUM

Después de obtener los requerimientos y tener una visión del maquetado de interfaces, se procedió a realizar el desarrollo SCRUM, iniciando con la definición del rol de usuario y la elaboración de las historias de usuario.

3.3.1 Rol de usuario e historias de usuario

Para este caso en particular, al ser un sistema de uso exclusivo, el único rol a considerar es el de **coordinador de carrera**.

Dentro de las historias de usuario, se muestran bajo el formato “Yo como (rol) necesito (funcionalidad) para (beneficio)” con las características INVEST:

- Independent (Independientes).
- Negotiable (Negociable).
- Valuable (Valorable).
- Estimable (Estimable).
- Small, scalable, sized (Pequeña).
- Testable (Verificable).

Basado en lo anterior, se presentan las historias de usuario para este caso en específico:

ID # C-1

Título: Inicio de sesión

Yo como: Coordinador
Necesito: Ingresar al sistema
Para: Poder administrar la información de la carrera

Figura 19. Historia de Usuario - Inicio de Sesión

ID # C-2

Título: Visualizar informe general

Yo como: Coordinador
Necesito: Ver la información general de ITI
Para: Conocer el estado general de la carrera con respecto a materias y estudiantes.

Figura 20. Historia de Usuario - Visualizar informe general

ID # C-3

Título: Registrar un cardex

Yo como: Coordinador
Necesito: Registrar un cardex nuevo
Para: Poder almacenarlo en la base de datos

Figura 21. Historia de Usuario - Registrar un cardex

ID # C-4

Título: Actualizar un cardex

Yo como: Coordinador
Necesito: Actualizar un cardex existente
Para: Mantener al día la información del estudiante en cuestión

Figura 22. Historia de Usuario - Actualizar un cardex

ID # C-5

Título: Visualizar mapa gráfico

Yo como: Coordinador
Necesito: Ver el mapa gráfico del estudiante
Para: Conocer su avance dentro de la carrera

Figura 23. Historia de Usuario - Visualizar mapa gráfico

ID # C-6

Título: Consultar generación

Yo como: Coordinador
Necesito: Ver los estudiantes registrados en una generacion.
Para: Revisar la lista existente de estudiantes inscritos.

Figura 24. Historia de Usuario - Consultar generación

ID # C-7

Título: Buscar Estudiante

Yo como: Coordinador
Necesito: Buscar la informacion de un estudiante
Para: Saber su estado académico y avance en la carrera

Figura 25. Historia de Usuario - Buscar estudiante

ID # C-8

Título: Crear reporte por materia

Yo como: Coordinador
Necesito: Realizar un reporte de una materia comparando dos periodos
Para: Poder observar el comportamiento de los datos

Figura 26. Historia de Usuario - Crear reporte por materia

ID # C-9

Título: Crear reporte por corte generacional

Yo como: Coordinador
Necesito: Realizar un reporte de una generacion en especifico
Para: Poder observar el avance de los estudiantes en general

Figura 27. Historia de Usuario - Crear reporte por corte generacional

ID # C-10

Título: Exportar reporte

Yo como: Coordinador
Necesito: Exportar mi reporte a excel
Para: Poder tener el reporte disponible sin necesidad de conectarme al sistema

Figura 28. Historia de Usuario - Exportar reporte

ID # C-11

Título: Mostrar graficas

Yo como: Coordinador
Necesito: Ver gráficas de los datos de la carrera
Para: Poder analizar de forma más fácil y rápida la información que solicito

Figura 29. Historia de Usuario - Mostrar gráficas

3.3.2 Estimación de Historias de Usuario

Es necesario realizar una estimación de las historias de usuario como una manera de dimensionar y relacionar el grado de complejidad entre estas. Para este fin, se valió de una técnica conocida como Planning Poker.

Esta técnica ágil consiste en que se tiene una baraja de cartas en donde los números se acomodan (con el previo acuerdo del equipo) para que así correspondan a la sucesión de Fibonacci, entonces se siguen los siguientes pasos:

1. El product owner presenta una historia de usuario.
2. Todos los participantes realizan su estimación, (sin influenciar al resto de los compañeros) y ponen su carta elegida boca abajo.

3. Una vez que todos hayan colocado sus cartas, se le da la vuelta y se observa la ponderación que cada uno otorgó, poniendo especial atención y discutiendo el porqué de dicha ponderación.
4. Después de la discusión se vuelve a estimar, esta vez con más información que no se tenía anteriormente.
5. El proceso termina hasta que se logra el consenso del equipo y luego se continúa con una nueva historia de usuario.

En la siguiente tabla (tabla 5) se presentan un total de 11 historias con sus respectivas ponderaciones obtenidas mediante la técnica que se explicó anteriormente.

Tabla 5. Historias de usuario

Historias de Usuario	
Ítem	Valoración en puntos
Iniciar Sesión	1
Visualizar información general	5
Registrar un cardex	8
Actualizar un cardex	8
Visualizar mapa gráfico	13
Consultar generación	1
Buscar Estudiante	3
Crear reporte por materia	13
Crear reporte por corte generacional	13
Exportar reporte	8
Mostrar gráficas	5

3.3.3 Product Backlog

El Product backlog es una lista ordenada de todo el trabajo pendiente, contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas según su valor para el negocio. Una vez terminado el paso anterior se procede a listar las historias de usuarios que se presenta a continuación en la tabla 6.

Tabla 6. Product Backlog

ID	Ítem	Tipo	Estado	Estimación
#C1	Iniciar Sesión	Historia de Usuario	No Iniciada	1
#C2	Visualizar información general	Historia de Usuario	No Iniciada	5
#C3	Registrar un cardex	Historia de Usuario	No Iniciada	8

#C4	Actualizar un cardex	Historia de Usuario	No Iniciada	8
#C5	Visualizar mapa gráfico	Historia de Usuario	No Iniciada	13
#C6	Consultar generación	Historia de Usuario	No Iniciada	1
#C7	Buscar Estudiante	Historia de Usuario	No Iniciada	3
#C8	Crear reporte por materia	Historia de Usuario	No Iniciada	13
#C9	Crear reporte por corte generacional	Historia de Usuario	No Iniciada	13
#C10	Exportar reporte	Historia de Usuario	No Iniciada	8
#C11	Mostrar gráficas	Historia de Usuario	No Iniciada	5

3.3.4 Plan de entregas

Una vez creado el product backlog es necesario definir y planificar los sprints o iteraciones. Se debe de tomar en cuenta que cada iteración debe de proporcionar un resultado, un incremento que pueda mostrarse cuando el cliente así lo solicite. Las puntuaciones son de mucha utilidad, debido a que, para cada sprint se debe realizar cierta cantidad de puntos, esto con el fin de evitar retrasos, lograr un entregable tangible en un tiempo determinado.

Para este caso se planearon 6 sprints, el total de puntos son 78, dando un total de 13 puntos por sprint (tabla 7 a tabla 12), se toma en cuenta que pueden haber más sprints, ya sea para probar módulos o funcionalidades que surgieron en las retroalimentaciones con el coordinador.

Tabla 7. Sprint 1

Sprint 1		
ID	Ítem	Estimación
#C1	Iniciar Sesión	1
#C3	Registrar un cardex	8
#C6	Consultar generación	1
#C7	Buscar estudiante	3
Total de puntos:		13

Tabla 8. Sprint 2

Sprint 2		
ID	Ítem	Estimación

#C2	Visualizar información general	5
#C4	Actualizar un cardex	8
Total de puntos:		13

Tabla 9. Sprint 3

Sprint 3		
ID	Ítem	Estimación
#C5	Visualizar mapa gráfico	13
Total de puntos:		13

Tabla 10. Sprint 4

Sprint 4		
ID	Ítem	Estimación
#C8	Crear reporte por materia	13
Total de puntos:		13

Tabla 11. Sprint 5

Sprint 5		
ID	Ítem	Estimación
#C9	Crear reporte por corte generacional	13
Total de puntos:		13

Tabla 12. Sprint 6

Sprint 6		
ID	Ítem	Estimación
#C10	Exportar reporte	8
#C11	Mostrar gráficas	5
Total de puntos:		13

Capítulo IV. Diseño de la base de datos

4.1 Diseño lógico

El diseño de la base de datos para este caso en especial se basó en el diagrama de clases que se presentó anteriormente, cabe recordar que el diagrama de clases representa a los objetos y sus relaciones en un nivel diferente y no todas pasan de forma transparente cuando se orienta a la base de datos.

A pesar de que el concepto sea el mismo, en el diagrama de clases se manejan objetos y relaciones que, no necesariamente se necesitan en la base de

datos para que esta funcione y complemente al sistema. A continuación, se presentan el Modelo Entidad – Relación y el diccionario de datos para este caso.

4.1.1 Modelo Entidad Relación

Una buena práctica en bases de datos es utilizar el Modelo Entidad-Relación (MER) para representar de forma simple la interacción de las entidades y sus atributos. Definamos como:

- **Entidad:** Es un objeto o cosa del mundo real que se diferencia de otras por sus atributos.
- **Atributo:** Son las características que definen a una entidad.
- **Relación:** Es la interacción que tendrá con otras entidades, (establecido por un atributo en concreto conocido como llave).

A continuación, en la figura N, se presenta el MER que se usará de base para realizar más adelante una forma extendida, es decir, la forma en la cual ya se presenta la base de forma concreta y lista para su implementación.

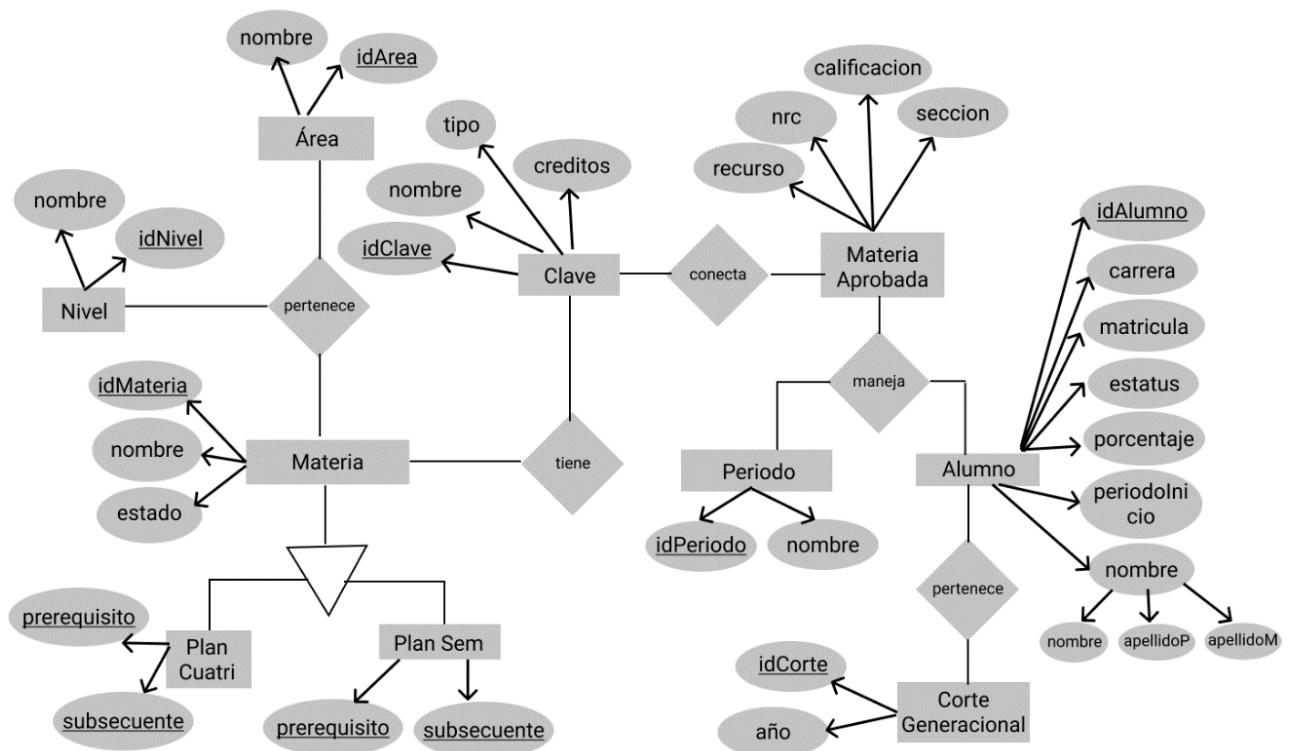


Figura 30. Modelo Entidad Relación

4.1.2 Diccionario de datos

Se creó el diccionario de datos (DDD) para observar los datos que el sistema va a tener y poder entender cada uno ya que, permite definir su significado, tipo y relaciones con las demás entidades.

Las tablas 13 a 23 muestran los diferentes datos que tendrán las entidades presentes en la base de datos.

Área

Tabla 13. DDD de entidad Área

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idArea (<i>Primaria</i>)	Identificador del área	int	No	-
NombreArea	Nombre del área del conocimiento	varchar(45)	No	-

Nivel

Tabla 14. DDD de entidad Nivel

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idNivel (<i>Primaria</i>)	Identificador del nivel	int	No	-
nombreNivel	Nombre del nivel	varchar(11)	No	-

Periodo

Tabla 15. DDD de entidad Periodo

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idPeriodo (<i>Primaria</i>)	Identificador del periodo	int	No	-
nombrePeriodo	Nombre del periodo escolar	varchar(45)	No	-

Materia

Tabla 16. DDD de entidad Materia

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idMateria (Primaria)	Identificador de la materia	int	No	-
nombreMateria	Nombre de la materia	varchar(45)	No	-
idArea (Foránea)	Identificador del área al que pertenece	int	Si	Area
idNivel (Foránea)	Identificador del nivel al que pertenece	int	No	Nivel

Clave

Tabla 17. DDD de entidad Clave

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idClave (Primaria)	Identificador de la clave	int	No	-
nombreClave	Clave de la materia	varchar(10)	No	-
creditos	Cantidad de créditos	int	No	-
tipo	Bandera para saber si es clave semestral o cuatrimestral	varchar(2)	No	-
idMateria (Foránea)	Identificador de la materia a la que pertenece	int	No	Materia

Alumno

Tabla 18. DDD de entidad Alumno

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idAlumno (Primaria)	Identificador del estudiante	int	No	-

matricula	Matricula del estudiante	varchar(9)	No	-
nombre	Nombre con apellidos del estudiante	varchar(60)	No	-
estatus	Estatus del estudiante	varchar(25)	No	-
periodoInicio	Periodo de ingreso del estudiante	varchar(45)	No	-
carrera	Nombre de la carrera que cursa el estudiante	varchar(45)	No	-
porcentaje	Porcentaje de avance de carrera	int	No	-
idCorte (Foránea)	Identificador del corte generacional al que pertenece	int	No	Corte_generacional

Materia Aprobada

Tabla 19. DDD de entidad Materia Aprobada

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idAlumno (Primaria)	Identificador del estudiante	int	No	Alumno
idClave (Primaria)	Identificador de la clave	int	No	Clave
idPeriodo	Identificador del periodo en el que se aprobó la materia	int	No	Periodo
nrc	NRC de la materia	int	No	-
calificacion	Calificación obtenida	varchar(2)	No	-
recurso	Recurso aplicado a la materia	varchar(10)	Si	-

seccion	Sección donde se cursó la materia	varchar(11)	No	-
----------------	-----------------------------------	-------------	----	---

Corte Generacional

Tabla 20. DDD de entidad Corte Generacional

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idCorte	Identificador del corte generacional	int	No	-
agnoGeneracion	Año de la creación de la generación	varchar(4)	No	-

Administrador

Tabla 21. DDD de entidad Administrador

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idAdministrador	Identificador del administrador	int	No	-
numTrabajador	Número de trabajador institucional del administrador	varchar(45)	No	-
nombre	Nombre del administrador	varchar(45)	No	-
apellidop	Apellido paterno del administrador	varchar(45)	No	-
apellidom	Apellido materno del administrador	varchar(45)	No	-
password	Contraseña de cuenta para el administrador	varchar(45)	No	-

Plan Cuatri

Tabla 22. DDD de entidad Plan Cuatrimestral

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
prerequisito (Primaria)	Identificador de la materia requisito	int	No	Materia
subsecuente (Primaria)	Identificador de la materia subsecuente	int	No	Materia

Plan Sem

Tabla 23. DDD de entidad Plan Semestral

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
prerequisito (Primaria)	Identificador de la materia requisito	int	No	Materia
subsecuente (Primaria)	Identificador de la materia subsecuente	int	No	Materia

4.2 Diseño Físico

Una vez realizado el DDD y el MER, se procedió a realizar el MER extendido, que consistió en la creación física de la base de datos a través de un SGBD (Sistema Gestor de Bases de Datos), esto se logra trabajando con el lenguaje SQL.

4.2.1 Modelo Entidad Relación Extendido

Para lograr la creación y la manipulación necesaria para trabajar lo anterior, se usó MySQL Workbench, que es una herramienta que permite manejar de forma visual las bases de datos.

A continuación, en la figura 31, se muestra el diagrama realizado en Workbench para este sistema.

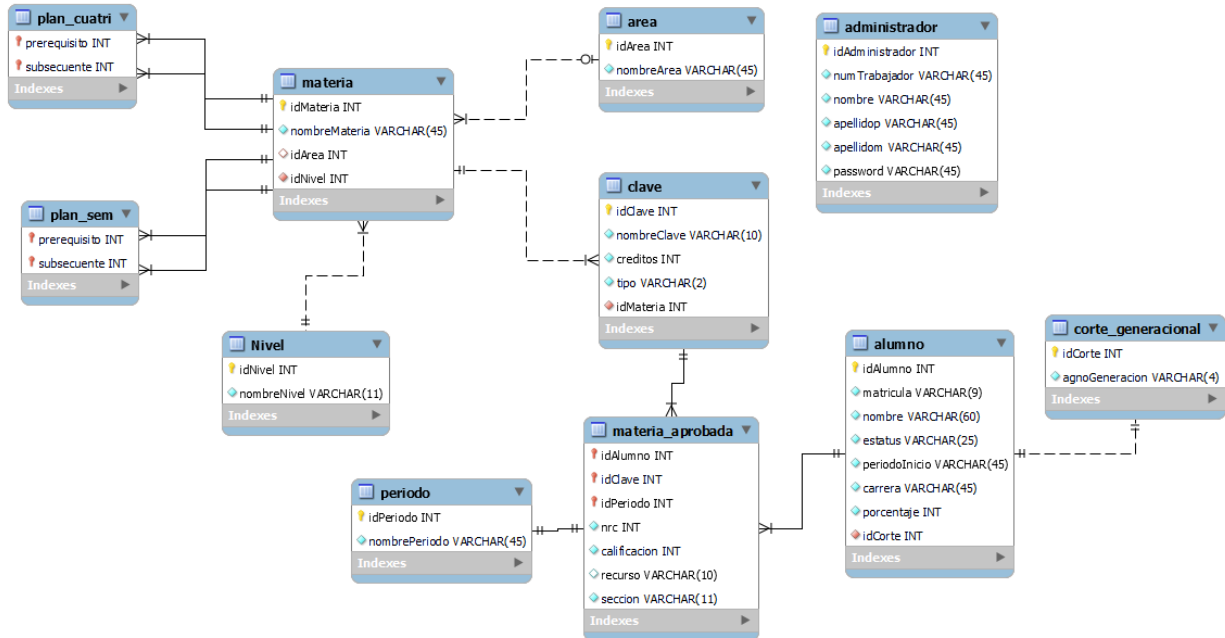


Figura 31. Modelo Entidad Relación Extendido con Workbench

Un detalle importante que se observa dentro de la base de datos son las tablas **plan_cuatri** y **plan_sem**, a pesar de que, tienen la misma estructura se manejan de forma separada debido a que en el manejo y conexión de materias en los planes las materias interactúan de formas diferentes, como por ejemplo la cantidad de materias con las que conectan, materias de nueva creación dentro del plan semestral y que no existen en el plan cuatrimestral o viceversa.

4.3 Código SQL

Una de las ventajas de MySQL Workbench es que permite transformar el diagrama anterior en código SQL, se hizo uso de esta funcionalidad de forma parcial, es decir, solo se obtuvo la plantilla de las tablas para su creación, mas no para la inserción de los datos ni las consultas dentro de éstas.

El código SQL se encuentra en el anexo 1 dentro de la sección de ANEXOS.

Capítulo V. Implementación y pruebas

5.1 Implementación

En este apartado se mostrará el resultado del sistema web, con base al proceso que se ha llevado anteriormente.

El framework utilizado para crear esta aplicación fue CodeIgniter en su versión 4, se instaló la biblioteca Bootstrap para el diseño de las interfaces las cuales son responsivas y la biblioteca de Grocery Crud para apoyar la implementación con respecto a la gestión del módulo de reportes de la aplicación.

En cuestiones más específicas de la interfaz, al ser software institucional, se dicta una paleta de colores en tonos azules y blancos. Las imágenes son de una biblioteca open source y las gráficas están hechas con la librería Chart.js. A continuación, se muestra la implementación que se realiza con lo que se mencionó anteriormente.

5.1.1 Inicio de Sesión

En esta sección, el coordinador debe colocar su numero de trabajador y la contraseña (que en su primera vez va a ser su numero de trabajador, después se debe actualizar), esto para acceder al sistema. En la figura 32, se muestra la interfaz en despliegue del inicio de sesión.

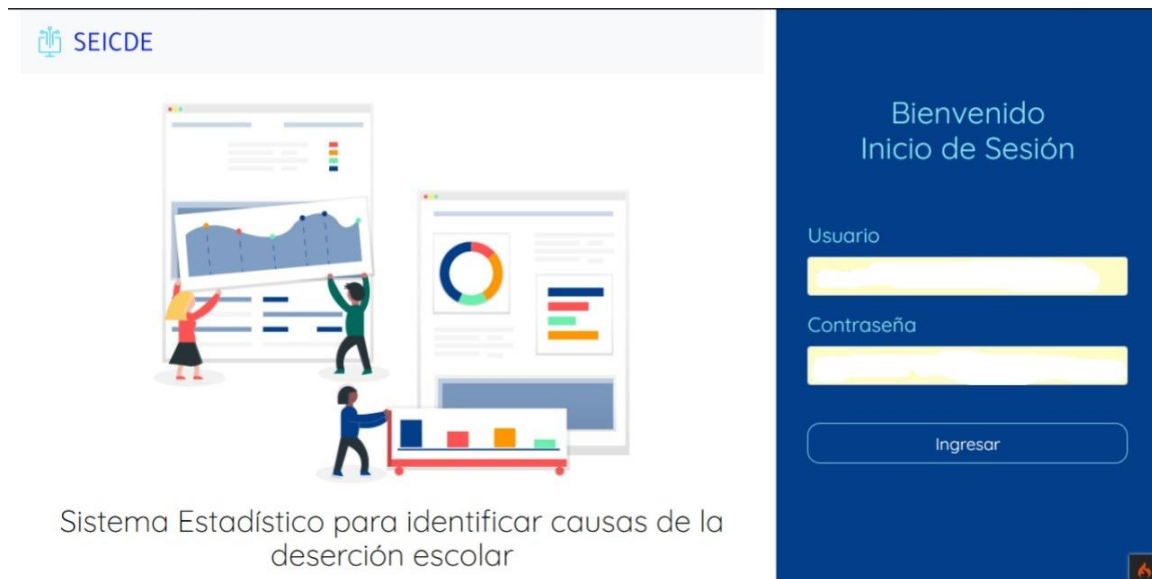


Figura 32. Inicio de Sesión

5.1.2 Informe principal de carrera

Al acceder, el coordinador observará un informe generalizado del estado de la carrera, que incluye los estudiantes registrados (en el sistema), promedio general de esta, materias registradas y materias activas (que están cursando o han cursado, por la apertura de optativas), junto con gráficas de las materias con los promedios más altos y bajos, esto va a variar de acuerdo con los registros de las calificaciones de todos los estudiantes (figura 33 a 34).

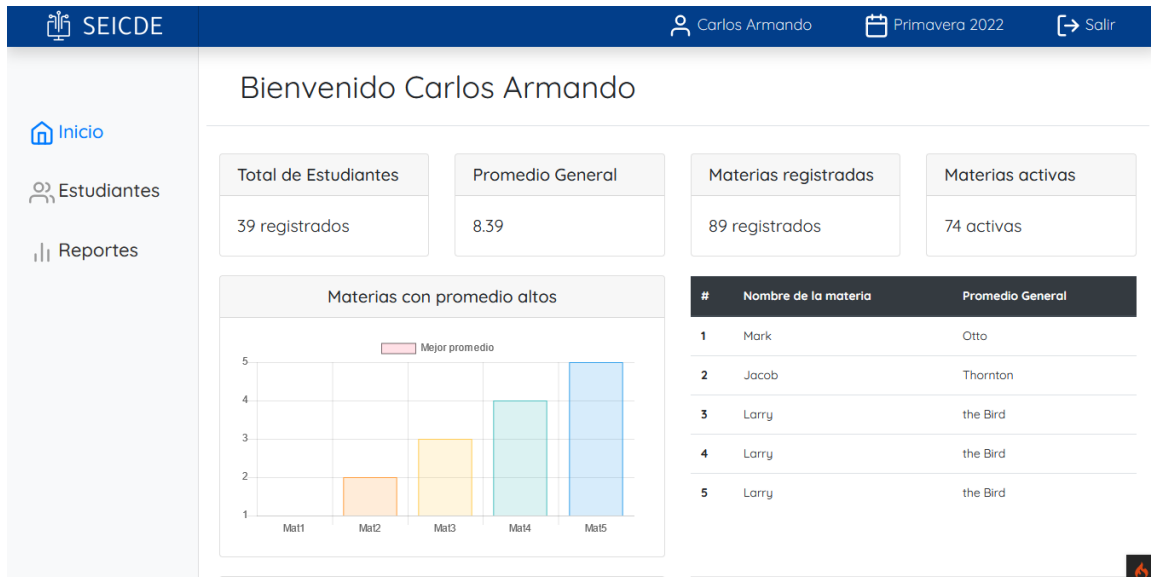


Figura 33. Informe principal

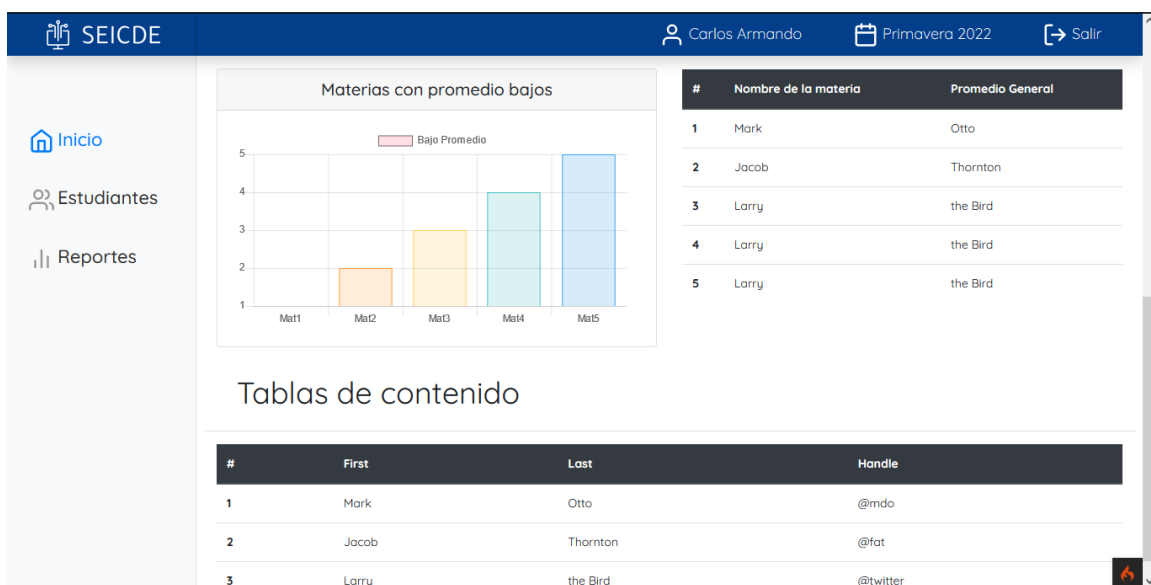


Figura 34. Informe principal (parte 2)

5.1.3 Cortes Generacionales y Mapas Gráficos

En la sección de estudiantes, el coordinador visualizará las generaciones que han sido registradas en el sistema, cada una está agrupada por año (figura 35), al entrar a una de estas, se desplegará una lista de estudiantes que integran dicha generación con los siguientes datos: nombre completo, matricula, estatus, porcentaje de avance, numero de materias aprobadas y un botón que permite acceder al mapa gráfico (figura 36).



Figura 35. Cortes Generacionales



Figura 36. Lista de estudiantes de una generación

El coordinador visualizará un mapa donde hay cuadros que contienen las materias, automáticamente se cargara un marcado con palomita verde para las aprobadas, de advertencia para materias que no ha cursado y de una equis roja, que marca aquellas en las cuales no se tiene derecho (figura 37 a 38).

Existen dos mapas, un cuatrimestral y un semestral, puesto que, los planes son diferentes y manejan sus propias restricciones.

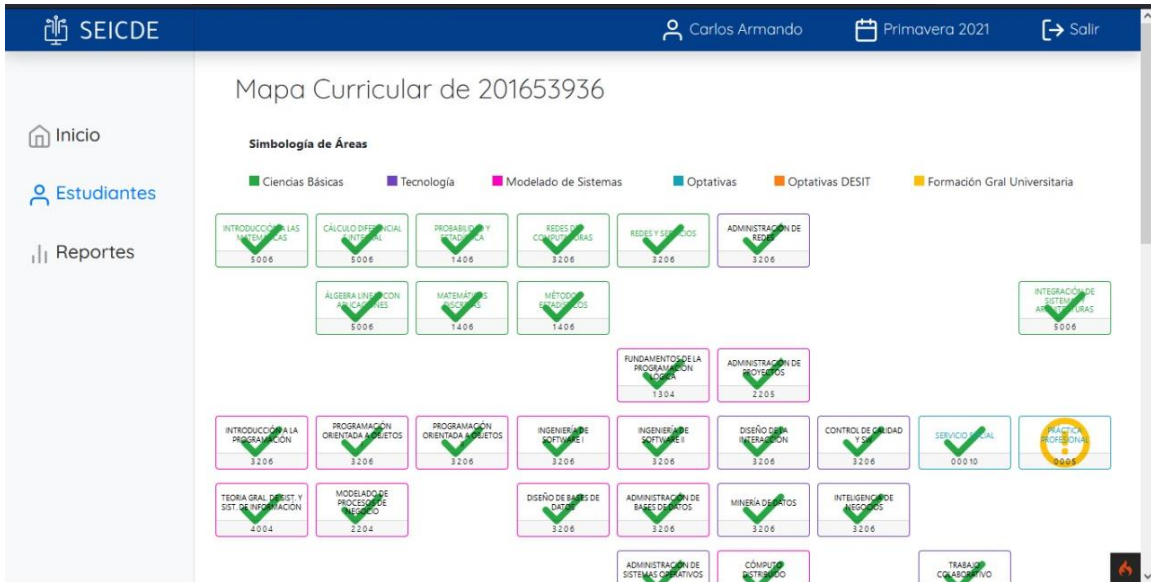


Figura 37. Mapa gráfico de un estudiante



Figura 38. Sección mapa gráfico para optativas

5.1.4 Reportes por Materia

Dentro de la sección de reportes se encontrarán dos opciones: reportes por materia y por corte generacional (figura 39). En el caso de materia, se solicitarán los siguientes datos: materia y dos periodos para comparar el comportamiento de estos con respecto a la materia. En el despliegue, se mostrará dos graficas correspondiente a los periodos y tablas con información de estas (figuras 40 a 43).



Figura 39. Menú para reportes

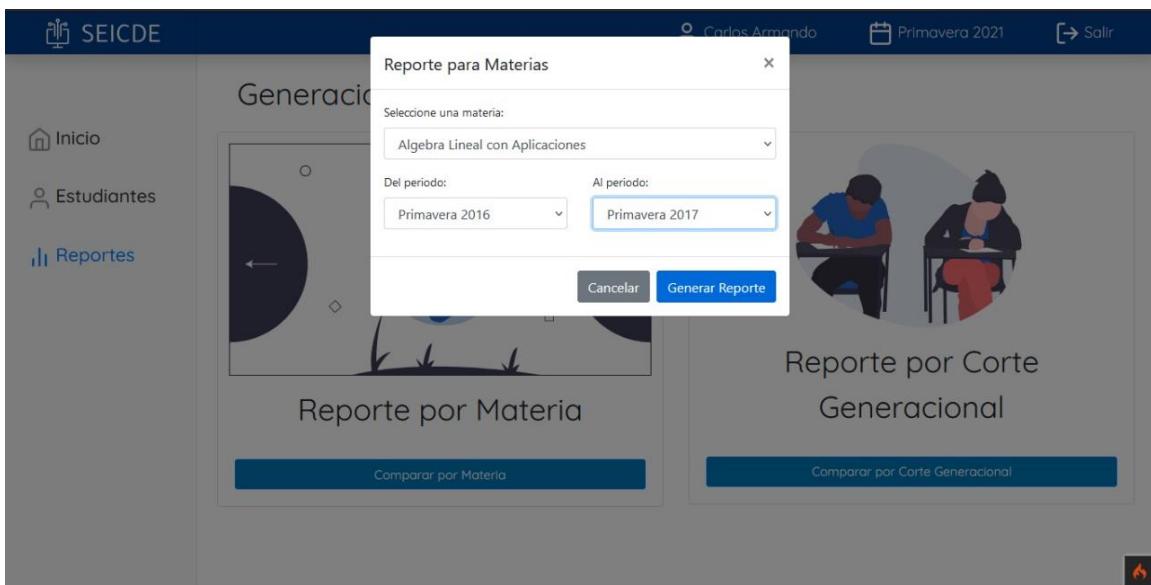


Figura 40. Reporte por materia

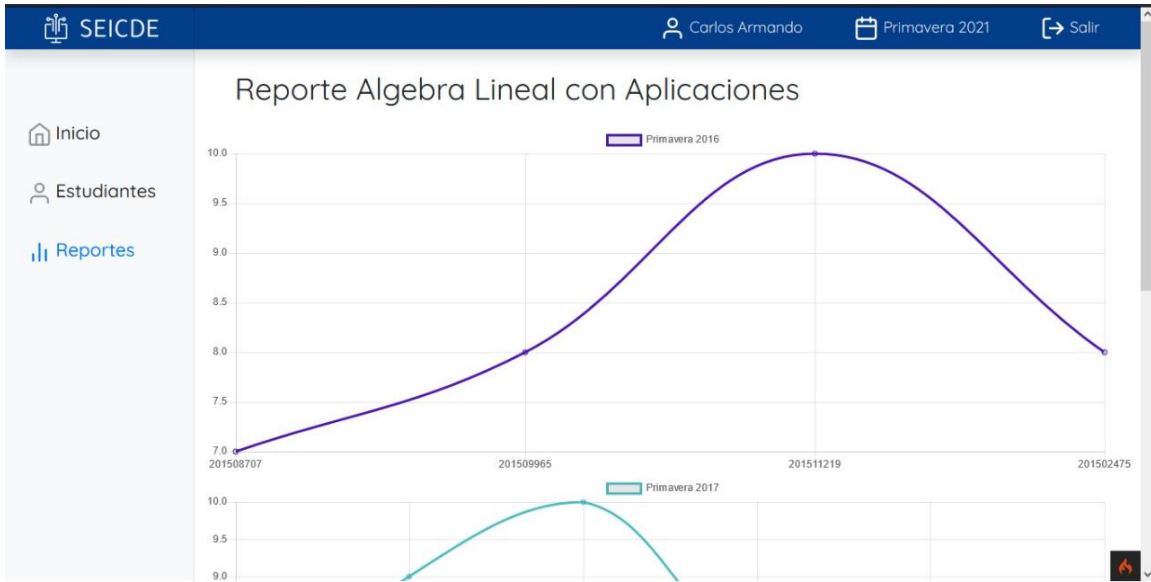


Figura 41. Reporte de una materia (primera gráfica)



Figura 42. Reporte de una materia (segunda gráfica)

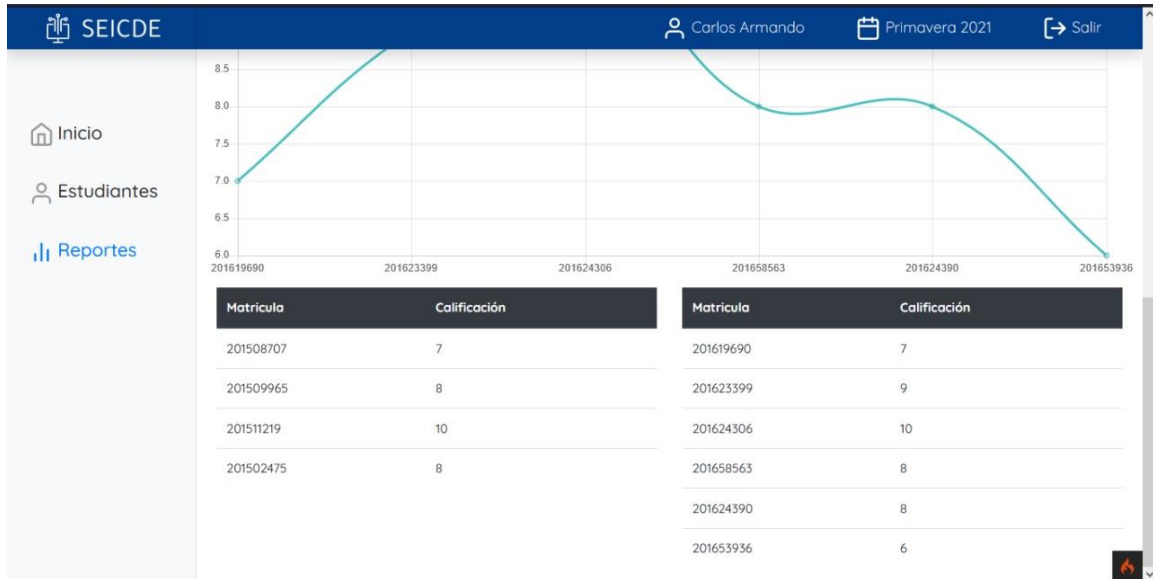


Figura 43. Tablas de datos para reporte

5.1.5 Reportes por Corte generacional

En el caso de los cortes generacionales, se solicitará el año de la generación que quiere visualizar. En el despliegue, se mostrará la información general de la generación como lo son estudiantes registrados, promedio general, promedio más alto y bajo, así como una gráfica que desplegará información sobre las materias durante un periodo específico dentro de la generación como número de estudiantes que cursaron una materia. (figuras 44 a 46).

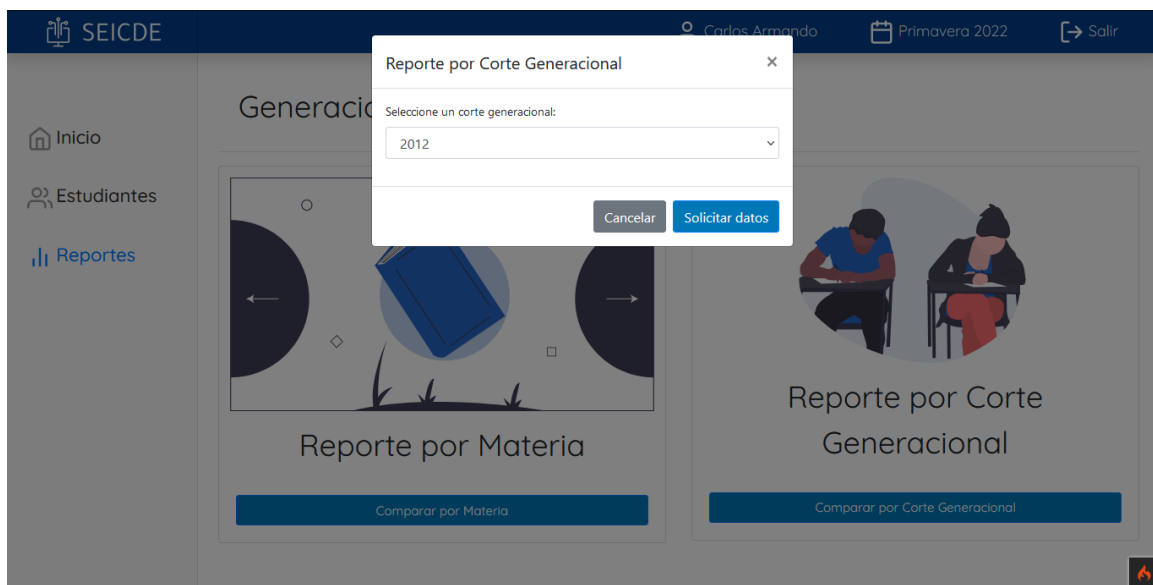


Figura 44. Reporte por corte generacional

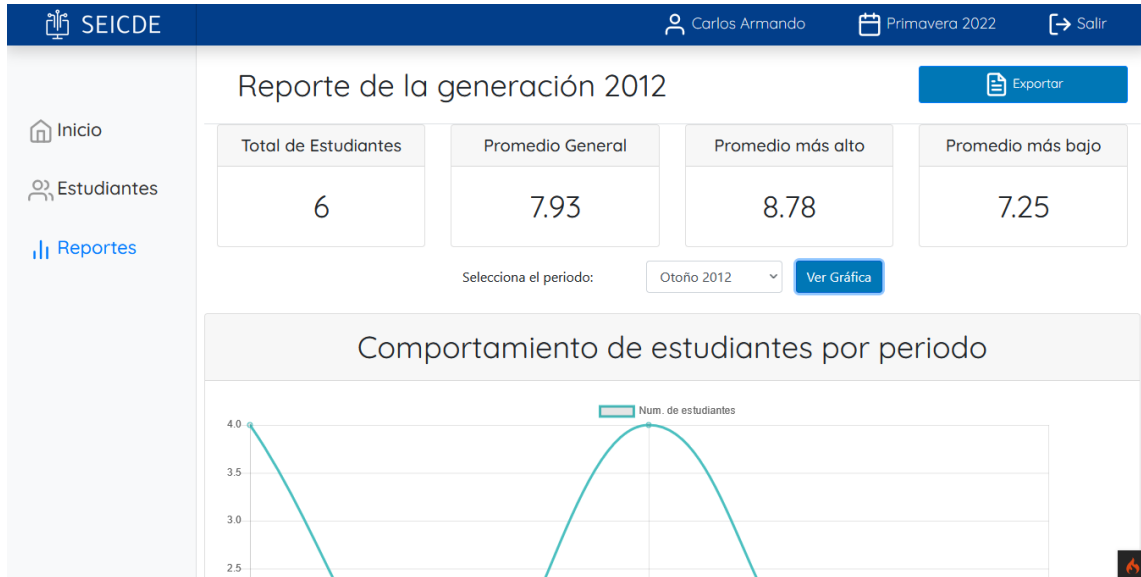


Figura 45. Reporte por corte generacional (despliegue)

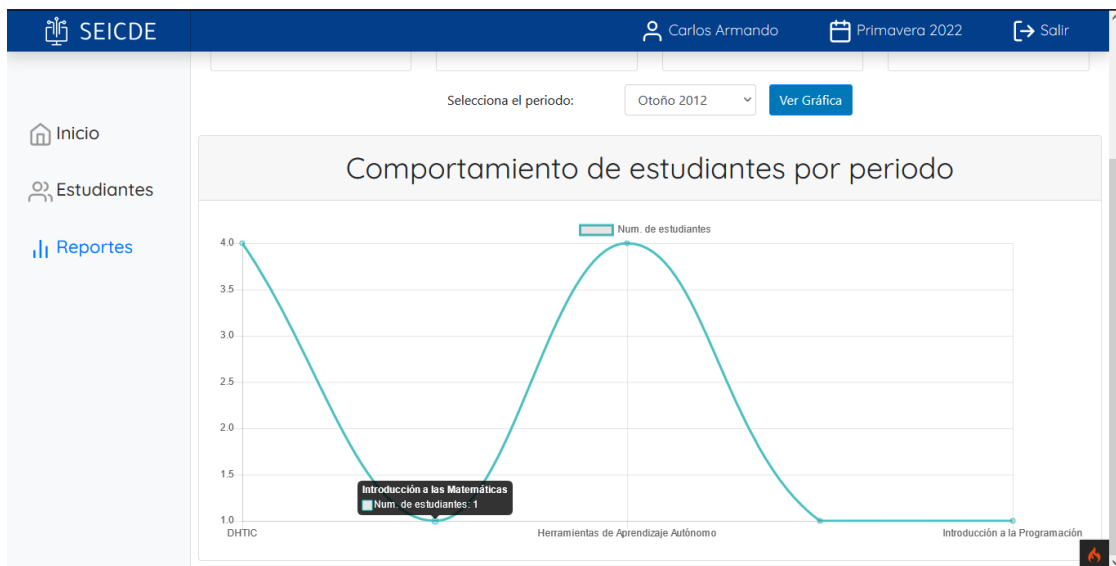


Figura 46. Reporte por corte generacional (vista de gráfica)

5.2 Pruebas

Dentro del desarrollo del sistema se contempló el concepto de calidad y para lograr esto, se implementó dos tipos de pruebas: caja negra y unitarias. A continuación, se muestran las pruebas realizadas y sus respectivos resultados.

5.2.1 Pruebas Caja Negra

Cuando se habla de pruebas de caja negra se habla de aquellas pruebas en las cuales se toma en cuenta las entradas y salidas requeridas, sin tomar en cuenta el funcionamiento mediante código del sistema.

En la tabla 24 se muestran las pruebas realizadas en este estilo, siendo caso positivo y, en caso de existir, la contraparte, para entender donde se puede dar una falla.

Tabla 24. Pruebas de caja negra

Código	Requisitos	Entrada	Descripción	Salida	¿La prueba pasa?
PCN1	Consultar información de una generación	-Año de la generación a consultar	El coordinador da clic en el botón 'Ver generación' que desea consultar.	Se muestra la siguiente información: - Lista de estudiantes con: -Matricula -Nombre Completo -Estatus -Porcentaje -Materias Aprobadas -Botón para acceder al cardex	Sí
PCN2	Consultar información de una generación	-Año de la generación a consultar, (sin estudiantes registrados).	El coordinador da clic en el botón 'Ver generación' que desea consultar, pero internamente no hay estudiantes registrados.	Mensaje: No se encuentran estudiantes registrados en esta generación	Sí
PCN3	Buscar estudiante dentro de la generación	-Matricula o Nombre del estudiante	El coordinador ingresa una matrícula o un nombre dentro de una 'caja' de búsqueda.	Se muestra un estudiante con los siguientes datos: -icono -Matricula	Sí

				-Nombre completo -Estatus -Materias aprobadas -Botón para acceder a su cardex.	
PCN4	Buscar estudiante dentro de la generación	-Matricula con prefijo diferente o Nombre del estudiante no registrado	El coordinador ingresa una matrícula no existente/errónea o un nombre dentro de una 'caja' de búsqueda.	Mensaje: No se encuentra al estudiante.	No
PCN5	Generar reporte por materia	-ID de la materia -Nombre de la materia -Periodo escolar 1 -Periodo escolar 2	El coordinador ingresa al menú de reportes, da clic en 'Generar Reportes por Materia', selecciona la materia y los periodos de comparación.	Se cargan gráficas en blanco y las tablas de algunos de los periodos marcan que no cuentan con registros.	No
PCN6	Generar reporte por materia	-ID de la materia -Nombre de la materia -Periodo escolar 1 -Periodo escolar 2	El coordinador ingresa al menú de reportes, da clic en 'Generar Reportes por Materia', selecciona la materia y los periodos de comparación.	Se muestran dos gráficas lineales sobre los periodos solicitados y tablas con los datos.	Sí
PCN7	Generar reporte por corte generacional	-ID de la generación -Año de la generación	El coordinador ingresa al menú de reportes, da clic en 'Generar	Se muestran los siguientes datos: -Promedio general de la generación	Sí

			Reportes por Corte generacional', selecciona la generación y da clic en 'Generar'.	-Número de estudiantes registrados. -Materias con bajo desempeño. -Materias con alto desempeño	
PCN8	Generar Reporte por corte generacional	-ID erróneo de la generación -Año erróneo de la generación	El coordinador ingresa al menú de reportes, da clic en 'Generar Reportes por Corte generacional', selecciona la generación y da clic en 'Generar'.	No se carga ningún datos y envía un mensaje: No se ha encontrado información de la generación que ingresó.	No
PCN9	Añadir cardex	-Archivo en formato PDF/HTML	El coordinador coloca el cardex en una carpeta iniciando con la matricula del estudiante.	Carga la información en la base de datos y lo coloca en la lista de estudiantes que corresponda.	Sí
PCN10	Mostrar mapa gráfico de estudiante	-Matricula del estudiante	Una vez que el coordinador ha encontrado al estudiante, puede acceder a su mapa gráfico a través del botón 'Ver mapa gráfico'.	Se muestra el mapa gráfico, con las siguientes marcas: -Palomita color verde para materias aprobadas. -Advertencia de color amarillo para materias que faltan por cursar.	Sí

				- Equis color rojo, indicando materias a las que no tiene derecho	
PCN11	Mostrar mapa gráfico de estudiante	-Matricula errónea del estudiante (el prefijo no es el correcto)	Una vez que el coordinador ha encontrado al estudiante, puede acceder a su mapa gráfico a través del botón 'Ver mapa gráfico'.	Se muestra el mapa gráfico, pero sin marcas.	No

5.2.2 Pruebas Unitarias

Como parte de las pruebas, se realizaron algunas de tipo unitario debido a que se debe verificar el funcionamiento de las funciones escritas en el lenguaje JavaScript. Para ello, se usó al framework de pruebas **Jest** siendo ejecutado en el entorno de **Node JS**. A continuación, se explica a detalle la configuración y estructura de la prueba en Jest.

Se inició un proyecto independiente del original (con su respectivo repositorio), para preservar la integridad del código y no combinar entornos y frameworks. En ese proyecto se instaló Jest a través de Node JS con **'npm install --save-dev jest'** (esto indica que se agrega la dependencia **jest**, y que se agrega para el ambiente de desarrollo **--save-dev**).

Una vez realizado lo anterior, en la carpeta de proyecto se trabaja de la siguiente forma (figura 47):

```
class Materias{}

//Se exporta la clase para poder usarla en la unit test
module.exports = Materias
```

Figura 47. Clase "piloto" en JavaScript

La clase Materias es la que va a contener las funciones que están en el proyecto original, se recomienda trabajar con POO en JavaScript para evitar errores en la ejecución de la prueba. En la última línea se maneja uno de los tipos de exportación por módulos en JS.

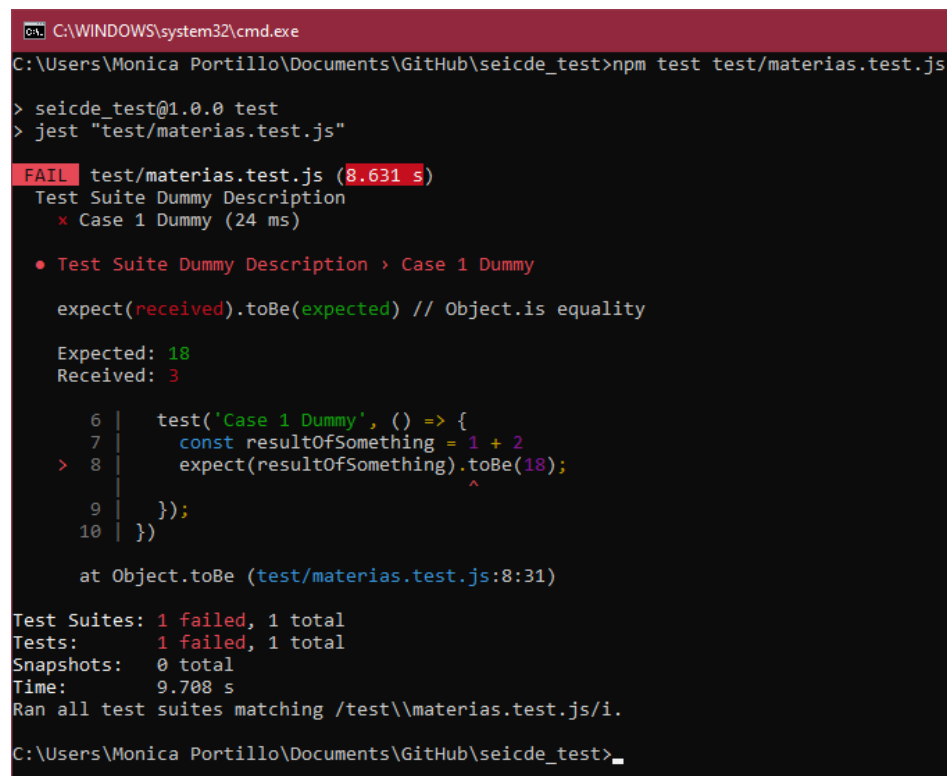
Por cada clase, también se recomienda tener un archivo con extensión **<archivo/clase_js>.test.js** para poder manejar las pruebas, como el que se muestra a continuación en la figura 48.

```
//se importa las clases necesarias para ejecutar las pruebas
const Materias = require('../app/materias')

//Se escribe la prueba
describe("Test Suite Dummy Description", () => {
  test('Case 1 Dummy', () => {
    const resultOfSomething = 1 + 2
    expect(resultOfSomething).toBe(18);
  });
});
```

Figura 48. Muestra de prueba "piloto" para trabajar con Jest

En el código se muestra a un objeto constante llamado Materias, el cuál va a llamar a la clase Materias que se definió anteriormente. Para esta prueba piloto no fue necesaria (pero se usó más adelante) solo se muestra el funcionamiento de Jest. En la ejecución dentro de Node se inicia a través de **'npm test <carpeta donde está la prueba>'** (figura 49).



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Monica Portillo\Documents\GitHub\seicde_test>npm test test/materias.test.js

> seicde_test@1.0.0 test
> jest "test/materias.test.js"

FAIL test/materias.test.js (8.631 s)
  Test Suite Dummy Description
    ✕ Case 1 Dummy (24 ms)

  ● Test Suite Dummy Description › Case 1 Dummy

    expect(received).toBe(expected) // Object.is equality

    Expected: 18
    Received: 3

       6 |   test('Case 1 Dummy', () => {
       7 |     const resultOfSomething = 1 + 2
    >    8 |     expect(resultOfSomething).toBe(18);
         |                                     ^
       9 |   });
      10 | })

      at Object.toBe (test/materias.test.js:8:31)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:  0 total
Time:        9.708 s
Ran all test suites matching /test\\materias.test.js/i.

C:\Users\Monica Portillo\Documents\GitHub\seicde_test>
```

Figura 49. Ejecución de prueba piloto con Jest en Node.js

Una vez entendido lo anterior, se muestran las pruebas que se ejecutaron en este caso en particular, se mostraran solo los resultados de las pruebas, el código que ejecutan las mismas se encontrará en el anexo 2.

Creación de un objeto Materia

La prueba consistió en la creación de un objeto Materia, esto con el fin de comprobar que, si se desea cargar la información que maneja el sistema a través de un formato ligero como lo es JSON (JavaScript Object Notation), puedan crearse de forma adecuada los objetos correspondientes (Figura 50).

```
> seicde_test@1.0.0 test
> jest "test/materias.test.js"
PASS test/materias.test.js (6.539 s)
  Pruebas para Materias
    ✓ 1.Creación de una materia (16 ms)
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        7.707 s, estimated 8 s
```

Figura 50. Prueba Unitaria para creación de materias

Ver plan semestral

Una de las funciones que más se ejecutan es la consulta de materias, sea una sola o en conjunto, esto representa una alta interacción en el intercambio de datos, en la figura 51 se muestra la prueba relacionada con lo mencionado anteriormente.

```
> seicde_test@1.0.0 test
> jest "test/materias.test.js"
PASS test/materias.test.js
  Pruebas para Materias
    ✓ 1.Creación de una materia (11 ms)
    ✓ 2.Ver Materias semestrales (17 ms)
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        5.332 s
```

Figura 51. Prueba Unitaria de visualización

Consultar una materia aprobada

Este es el primer paso para poder manejar la información contenida en el cardex del estudiante, por eso es vital saber si se logra hacer la búsqueda con las materias de su respectivo plan y con las registradas en la base de datos (figura 52).

```
> seicde_test@1.0.0 test
> jest "test/materia_aprobada.test.js"

console.log
  La materia es: ID: 46
  Nombre: Herramientas Web
  Calificacion:7

    at Function.log [as buscarMateriaAprobada] (app/materia_aprobada.js:326:11)
PASS test/materia_aprobada.test.js (5.622 s)
  Pruebas para Materias Aprobadas
    ✓ 1.Buscar materia aprobada (234 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        6.492 s
```

Figura 52. Prueba Unitaria de búsqueda de materia en cardex

Marcado de una materia en mapa gráfico

Esta función es una de las complejas dentro del sistema por el manejo de los planes, requisitos de cada materia y la identificación de cada materia, al ser más visual, se simuló el marcado con banderas enteras y booleanas. En la figura 53, se muestra el resultado de esta prueba.

```
> seicde_test@1.0.0 test
> jest "test/materia_aprobada.test.js"

console.log
  Marca verde en: ID: 46
  Nombre: Herramientas Web

    at log (app/materia_aprobada.js:1078:17)
    at Array.find (<anonymous>)

PASS test/materia_aprobada.test.js (8.654 s)
  Pruebas para Materias Aprobadas
    ✓ 2.Marca de materias en mapa gráfico (481 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        9.893 s
```

Figura 53. Prueba Unitaria de Marcado de materias con banderas

Capítulo VI. Conclusiones y trabajos futuros

6.1 Conclusiones

Como hemos visto anteriormente, la deserción escolar es un factor persistente en la población estudiantil mexicana y en la comunidad de ITI – BUAP no es la excepción. Debemos apoyar a los estudiantes desde un punto de acción académico, puesto que, a partir de este se observa el desempeño de cada uno de estos y con esto, un rendimiento a nivel general de la carrera con el apoyo de herramientas tecnológicas.

El uso de estas herramientas como lo son los sistemas web basados en frameworks y las librerías tipo open source pueden permitir este apoyo en un ángulo flexible y mantenible a lo largo del tiempo. SEICDE se desarrolló pensando en lo anterior, que permita trabajar con todos los datos que se disponen de los estudiantes para darles una utilidad que permita a la coordinación armar planes de acción que incluso se pueda extender a otras carreras.

En esta versión se maneja a ITI como referencia, así como el manejo de los planes académicos que se encuentran vigentes en la institución en un formato más simple y accesible, así como la generación de reportes en Excel, para tener disponible la información sin necesidad de conectarse al sistema.

Este sistema se encontrará en un repositorio privado de GitHub, debido a la información contenida y el uso de estos. También se hace la observación de que se debe complementar con un eje pedagógico para mejorar los métodos de enseñanza ya existentes.

6.2 Trabajo futuro

Dentro del trabajo futuro se puede abordar las siguientes perspectivas

API REST: dentro de esta API se podría contener los datos de los estudiantes y planes curriculares (materias y claves) para que pueda ser consumida desde cualquier aplicación y/o lenguaje.

Integración del profesorado a la base de datos: el cuerpo académico que imparte las clases que cursa el alumno en su momento, pueden ser de utilidad para saber el desempeño estudiantil que se maneja durante el periodo que imparte la materia, permitiría a tener más precisión sobre que trabajar dentro de una estrategia coordinador-profesorado.

Modelo Machine Learning: al ser una base que a lo largo del tiempo va a crecer de forma masiva, se pueden aprovechar algunos servicios o herramientas ML (como en el caso de proveedores de nube como Microsoft Azure o Google Cloud

Platform) para utilizar algún modelo que nos permita saber más sobre los datos o en un caso de crear alguno, poderlo probar con seguridad y capacidad de escalabilidad y elasticidad a lo largo del tiempo.

Protección de datos: temporalmente el sistema se encuentra en un estado “frágil” en el cuál puede sufrir un ataque para extraer la información, por eso se podría implementar cifrado del lado de la base y una autenticación de doble factor para el sistema en general.

Optimización a través de la integración con Power BI: muchas de las gráficas y de las vistas de los datos se hacen a través del frontend, haciendo que el sistema sufra un poco de retraso en la carga, para una optimización y haciendo uso de herramientas más sofisticadas con respecto a manejo de datos, se ve la posibilidad de una integración con Power BI, a través de un frame incrustado para observar los datos de una forma más compacta.

Referencias

1. Asociación Nacional de Universidades e Instituciones de Educación Superior ANUIES. (2019). Anuario Educación Superior-Licenciatura Ciclo escolar 2017-2018. Recuperado de <http://www.anui.es.mx/informacion-y-servicios/informacion-estadistica-de-educacion-superior/anuario-estadistico-de-educacion-superior>
2. Marqués, M. (2011). Base de datos. Recuperado de <https://libros.metabiblioteca.org/bitstream/001/353/5/978-84-693-0146-3.pdf>
3. López Pedraza, F. J., Macías Gonzalez, M. C., & Sandoval García, E. (2019). Minería de datos: identificando causas de deserción en las instituciones públicas de educación superior de México. TIES, Revista de Tecnología e Innovación en Educación Superior, 1(2), 2-12. Obtenido de <http://ties.unam.mx/>
4. Reyes Nava, A., Flores Fuentes, R., & Alejo, R. (2017). Minería de datos aplicada para la identificación de factores de riesgo en alumnos. Research in Computing Science(139), 177-189.
5. Valero Orea, S., Salvador Vargas, A., & García Alonso, M. (2010). Minería de datos: predicción de la deserción escolar mediante algoritmo de árboles de decisión y el algoritmo de k vecinos más cercanos. Recursos digitales para la educación y la cultura, Kaambal, 33-39.
6. Hernández C., J. (2015). MODELO DE MINERÍA DE DATOS PARA IDENTIFICACIÓN DE PATRONES QUE INFLUYEN EN EL APROVECHAMIENTO ACADÉMICO (tesis de maestría). Instituto Tecnológico de La Paz, Baja California Sur, México.

7. Marcano, Y. J., & Rodríguez, R. H. (2014). MINERÍA DE DATOS APLICADA A LA DESERCIÓN ESTUDIANTIL. CASO: LICENCIATURA EN COMPUTACIÓN-UNIVERSIDAD DEL ZULIA-NPF. EDUCARE, 18(2), 31-51.
8. Foundation, C. (7 de octubre de 2020). CodeIgniter. Obtenido de <https://codeigniter.com/>
9. Group, T. P. (7 de octubre de 2020). Manual de PHP. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
10. Skoumbourdis, J. (7 de octubre de 2020). Grocery CRUD. Obtenido de <https://www.grocerycrud.com/>
11. Balsamiq S. LLC. (27 de octubre de 2020). Balsamiq Wireframes. Obtenido de <https://balsamiq.com/wireframes/desktop/#>
12. Foundation, O. J. (01 de junio de 2022). Acerca de Node JS. Obtenido de <https://nodejs.org/es/about>
13. Source, F. O. (01 de junio de 2022). Jest: Delightful JavaScript Testing. Obtenido de <https://jestjs.io/es-ES/>

ANEXOS

Anexo 1. Código SQL

```
CREATE DATABASE `materias_iti`;
USE `materias_iti`;
```

```
CREATE TABLE `administrador` (
  `idAdministrador` INT NOT NULL AUTO_INCREMENT,
  `numTrabajador` VARCHAR(45) NOT NULL,
  `nombre` VARCHAR(45) NOT NULL,
  `apellidop` VARCHAR(45) NOT NULL,
  `apellidom` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idAdministrador`));
```

```
CREATE TABLE `area` (
```

```
`idArea` INT NOT NULL AUTO_INCREMENT,  
`nombreArea` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`idArea`));
```

```
CREATE TABLE `nivel` (  
  `idNivel` INT NOT NULL AUTO_INCREMENT,  
  `nombreNivel` VARCHAR(11) NOT NULL,  
  PRIMARY KEY (`idNivel`));
```

```
CREATE TABLE `materia` (  
  `idMateria` INT NOT NULL AUTO_INCREMENT,  
  `nombreMateria` VARCHAR(55) NOT NULL,  
  `idArea` INT NULL,  
  `idNivel` INT NOT NULL,  
  PRIMARY KEY (`idMateria`),  
  FOREIGN KEY (`idArea`) REFERENCES `area` (`idArea`),  
  FOREIGN KEY (`idNivel`) REFERENCES `nivel` (`idNivel`));
```

```
CREATE TABLE `corte_generacional` (  
  `idCorte` INT NOT NULL AUTO_INCREMENT,  
  `agnoGeneracion` VARCHAR(4) NOT NULL,  
  PRIMARY KEY (`idCorte`));
```

```
CREATE TABLE `periodo` (  
  `idPeriodo` INT NOT NULL AUTO_INCREMENT,  
  `nombrePeriodo` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idPeriodo`));
```

```
CREATE TABLE `clave` (  
  `idClave` INT NOT NULL AUTO_INCREMENT,  
  `nombreClave` VARCHAR(10) NOT NULL,  
  `creditos` INT NOT NULL,  
  `tipo` VARCHAR(2) NOT NULL,  
  `idMateria` INT NOT NULL,  
  PRIMARY KEY (`idClave`),  
  FOREIGN KEY (`idMateria`) REFERENCES `materia` (`idMateria`));
```

```

CREATE TABLE `plan_sem` (
  `prerequisito` INT NOT NULL,
  `subsecuente` INT NOT NULL,
  PRIMARY KEY (`prerequisito`, `subsecuente`),
  FOREIGN KEY (`prerequisito`) REFERENCES `materia` (`idMateria`),
  FOREIGN KEY (`subsecuente`) REFERENCES `materia` (`idMateria`));

CREATE TABLE `plan_cuatri` (
  `prerequisito` INT NOT NULL,
  `subsecuente` INT NOT NULL,
  PRIMARY KEY (`prerequisito`, `subsecuente`),
  FOREIGN KEY (`prerequisito`) REFERENCES `materia` (`idMateria`),
  FOREIGN KEY (`subsecuente`) REFERENCES `materia` (`idMateria`));

CREATE TABLE `alumno` (
  `idAlumno` INT NOT NULL AUTO_INCREMENT,
  `matricula` VARCHAR(9) NOT NULL,
  `nombre` VARCHAR(60) NOT NULL,
  `estatus` VARCHAR(25) NOT NULL,
  `periodoInicio` VARCHAR(45) NOT NULL,
  `carrera` VARCHAR(45) NOT NULL,
  `porcentaje` INT NOT NULL,
  `materiasAprobadas` int(11) NOT NULL,
  `idCorte` INT NOT NULL,
  PRIMARY KEY (`idAlumno`),
  FOREIGN KEY (`idCorte`) REFERENCES `corte_generacional` (`idCorte`));

CREATE TABLE `materia_aprobada` (
  `idAlumno` INT NOT NULL,
  `idClave` INT NOT NULL,
  `idPeriodo` INT NOT NULL,
  `nrc` INT NOT NULL,
  `calificacion` VARCHAR(5) NOT NULL,
  `recurso` VARCHAR(10) NULL,
  `seccion` VARCHAR(11) NOT NULL,
  PRIMARY KEY (`idAlumno`, `idClave`, `idPeriodo`),
  FOREIGN KEY (`idAlumno`) REFERENCES `alumno` (`idAlumno`),
  FOREIGN KEY (`idClave`) REFERENCES `clave` (`idClave`),
  FOREIGN KEY (`idPeriodo`) REFERENCES `periodo` (`idPeriodo`));

```

Anexo 2

Para las materias:

//Se importa las clases necesarias para ejecutar las pruebas

```
const Materias = require('./../app/materias')
```

```
describe("Pruebas para Materias", () => {  
  test('1.Creación de una materia', () => {  
    //se ingresa una materia ficticia de  
    //prueba para comprobar que la clase construye el objeto correctamente  
    const nuevaMateria = new Materias(90,'Cálculo Diferencial','ITIS 610',6,1)  
  
    expect(nuevaMateria.id).toBe(90);  
    expect(nuevaMateria.nombre).toBe('Cálculo Diferencial');  
    expect(nuevaMateria.clave).toBe('ITIS 610');  
    expect(nuevaMateria.creditos).toBe(6);  
    expect(nuevaMateria.idArea).toBe(1);  
  });  
  test('2.Ver Materias semestrales', () => {  
  
    const mat1= {"clave": "ITIS 004", "creditos": "6", "id": "3", "idArea": "1",  
"nombre": "Cálculo Diferencial e Integral"}  
    const mat2 = {"clave": "ITIS 011", "creditos": "6", "id": "46", "idArea": "3",  
"nombre": "Herramientas Web"}  
    const mat3 = {"clave": "ITIS 700", "creditos": "6", "id": "62", "idArea": "6",  
"nombre": "Desarrollo de Videojuegos"}  
  
    const ps = Materias.verMateriasSemestrales()  
  
    expect(ps).toContainEqual(mat1);  
    expect(ps).toContainEqual(mat2);  
    expect(ps).toContainEqual(mat3);  
  });  
})
```

Para materias aprobadas:

```
const MateriasAprobadas = require('./../app/materia_aprobada')
```

```
describe("Pruebas para Materias Aprobadas", () => {  
  test('1.Buscar materia aprobada', () => {
```

```

    const busqueda = MateriasAprobadas.buscarMateriaAprobada('Herramientas
Web')
    const mat_aprobada = {"calificacion": "7", "creditos": "6", "idArea": "3",
"idMateria": "46", "nombreMateria": "Herramientas Web"}
    expect(busqueda.idMateria).toContain(mat_aprobada.idMateria);
    expect(busqueda.nombreMateria).toContain(mat_aprobada.nombreMateria);
    expect(busqueda.calificacion).toContain(mat_aprobada.calificacion);
    expect(busqueda.creditos).toContain(mat_aprobada.creditos);
    expect(busqueda.idArea).toContain(mat_aprobada.idArea);
  });
  test('2.Marca de materias en mapa gráfico', () => {
    const cardex_test= {"calificacion": "7", "creditos": "6", "idArea": "3", "idMateria":
"46", "nombreMateria": "Herramientas Web"}
    const ps = MateriasAprobadas.simulaMarcaje(cardex_test)

    expect(ps).toBe(true);
  });
})

```