

Desarrollo de una aplicación móvil, para realizar seguimiento y desempeño de los syllabus del área disciplinar del programa tecnología en desarrollo de software de la Universitaria Agustiniana

Doly Marley Rodriguez Arevalo
Han Carlot Galvis Llanes

Universitaria Agustiniana
Facultad de Ingenierías
Programa de Tecnólogo en Desarrollo de Software
Bogotá, D.C.
2020

Desarrollo de una aplicación móvil, para realizar seguimiento y desempeño de los syllabus del área disciplinar del programa tecnología en desarrollo de software de la Universitaria Agustiniana

Doly Marley Rodriguez Arevalo
Han Carlot Galvis Llanes

Director
Mauricio Alonso Villalba

Trabajo de grado para optar al título de Tecnólogo en Desarrollo de Software

Universitaria Agustiniana
Facultad de Ingenierías
Programa de Tecnólogo en Desarrollo de Software
Bogotá, D.C.
2020

Resumen

El objetivo de este proyecto es desarrollar una aplicación móvil, para realizar seguimiento y desempeño de los syllabus del área disciplinar del programa de tecnología en desarrollo de software de la Universitaria Agustiniiana; considerando la importancia del cumplimiento del syllabus para la finalidad de los programas académicos, se estima que es de gran utilidad recopilar la información suministrada por los estudiantes, permitiendo generar controles y seguimiento por parte de ellos. Para el desarrollo de la aplicación se usa la metodología programación extrema (XP), realizando un diseño intuitivo y sencillo y manejando varias interacciones para los resultados finales obtenidos. El funcionamiento de la aplicación se realiza un muestreo entre los estudiantes del programa en la jornada de la noche; utilizando como herramienta las encuestas, para conocer los resultados de las pruebas de la aplicación. De las pruebas realizadas a los estudiantes y desarrolladores, se obtuvieron resultados positivos tanto en las pruebas de software, como las de usabilidad.

Palabras clave: syllabus, seguimiento, desempeño, aplicación móvil.

Abstract

The objective of this project is to develop a mobile application to monitor and perform syllabus in the disciplinary area of the technology program in software development at the Agustiniana University; Considering the importance of complying with the syllabus for the purpose of academic programs, it is estimated that it is very useful to collect the information provided by students, allowing them to generate controls and monitoring by them. For the development of the application, the extreme programming methodology (XP) is used, making an intuitive and simple design and managing several interactions for the final results obtained. The operation of the application is sampled among the students of the program in the evening; using the surveys as a tool, to know the results of the tests of the application. From the tests carried out on students and developers, positive results were obtained both in software tests and in usability tests.

Keywords: syllabus, monitoring, performance, mobile application.

Tabla de contenidos

Introducción.....	12
1. Título del proyecto de software.....	13
2. Planeación del proyecto	14
2.1. Objetivos del proyecto.....	14
2.1.1. Objetivo general.....	14
2.1.2. Objetivos específicos.....	14
2.2. Planteamiento del problema y/o necesidad	14
2.3. Alcance del proyecto	15
2.4. Metodología de desarrollo de software	15
3. Marco teórico y estado del arte	25
3.1. Marco teórico.....	25
3.1.2. Syllabus.....	25
3.1.3. Marco institucional.....	25
3.1.4. Aplicación móvil.....	25
3.1.5. Diseño modular.....	26
3.1.6. Base de datos.....	26
3.1.7. Android studio.....	26
3.1.8. Ionic.....	27
3.1.9. Php.....	27
3.1.11. Visual studio code.....	27
3.1.12. JavaScript.....	27
3.2. Estado del arte	28
3.2.1. Classroom live web.....	28
3.2.2. Sistema de seguimiento de syllabus.....	28
3.2.3. Herramienta web para instrumento de seguimiento de syllabus.....	28
3.2.4. Cibercolegios.....	29
4. Especificación de requisitos de software (IEEE 830)	30
4.1. Perspectiva del producto.....	30
4.2. Funcionalidad del producto	30
4.3. Características de los usuarios	30
4.4. Restricciones.....	31

4.5.	Suposiciones y dependencias.....	31
4.6.	Requisitos específicos.....	32
4.6.1.	Actores/roles.....	32
4.6.2.	Requisitos funcionales.....	33
4.6.3.	Diagrama de casos de uso.	35
4.6.4.	Especificaciones de los casos de uso.....	36
4.7.	Requisitos de rendimiento	37
4.8.	Restricciones de diseño	37
4.9.	Atributos del software del sistema.....	37
4.9.1.	Requisitos no funcionales.....	37
4.9.2.	Portabilidad.	39
4.9.3.	Disponibilidad.	39
5.	Diseño del software (ISO -12207-1).....	40
5.1.	Diseño de la arquitectura de software.....	40
5.1.1.	El diagrama de contexto del sistema.	40
5.1.2.	El diagrama del contenedor.....	40
5.1.3.	El diagrama de componentes.....	41
5.2.	Diseño detallado del software.....	41
5.2.1.	Diagrama de clases.....	41
5.2.2.	Diagrama de paquetes.	42
5.2.3.	Diagramas de despliegue.....	42
5.3.	Diseño de la interfaz.....	43
5.3.1.	Interfaz gráfica de usuario.....	43
5.3.2.	Interfaces de entrada.....	44
5.3.3.	Interfaces de salida.....	46
6.	Implementación.....	49
6.1.	Plataformas de desarrollo	49
6.2.	Base de datos	49
6.3.	Infraestructura de hardware y redes.....	49
7.	Pruebas del software.....	51
7.1.	Pruebas del software.....	51
7.2.	Pruebas de usabilidad	55

Conclusiones60

Recomendaciones61

Referencias62

Lista de tablas

Tabla 1. Historia de usuario 1	17
Tabla 2. Historia de usuario 2	18
Tabla 3. Historia de usuario 3	18
Tabla 4. Historia de usuario 4	18
Tabla 5. Historia de usuario 5	19
Tabla 6. Tarjeta CRC de Registro del estudiante	21
Tabla 7. Tarjeta CRC de gestión del syllabus	21
Tabla 8. Tarjeta CRC de la encuesta	21
Tabla 9. Tarjeta CRC de informes.....	22
Tabla 10. Características del usuario estudiante.....	30
Tabla 11. Administrativo Uniagustiniana del área disciplinar	31
Tabla 12. Administrador de la base datos	31
Tabla 13. Rol del estudiante	32
Tabla 14. Rol del personal administrativo Uniagustiniana del área disciplinar	32
Tabla 15. Rol del administrador de base de datos.....	32
Tabla 16. Requisitos funcionales de la base de datos.....	33
Tabla 17. Requisitos funcionales del seguimiento syllabus	33
Tabla 18. Requisitos funcionales del desempeño syllabus.....	34
Tabla 19. Requisitos funcionales de los informes sobre syllabus	34
Tabla 20. Caso de uso del estudiante.....	36
Tabla 21. Caso de uso del administrativo de la Uniagustiniana.....	36
Tabla 22. Requisitos no funcionales del interfaz del sistema.....	37

Tabla 23. Requisitos no funcionales de ayuda en el uso del sistema	37
Tabla 24. Requisitos no funcionales de mantenimiento	38
Tabla 25. Requisitos no funcionales de seguridad en información	38

Lista de figuras

Figura 1. Diagrama de Gantt del proyecto.	19
Figura 2. Mockup de inicio de sesión, visualización de asignaturas y syllabus de ellas.	20
Figura 3. Mockup de seguimiento y desempeño del syllabus y exportes.	20
Figura 4. Funcionalidad del producto.....	30
Figura 5. Caso de uso general creado por los autores.	35
Figura 6. Caso de uso específico: estudiante.....	35
Figura 7. Caso de uso específico: Administrativos Uniagustiniana del área disciplinar.....	36
Figura 8. Diagrama de contexto del sistema.	40
Figura 9. Diagrama del contenedor.	40
Figura 10. Diagrama de componentes, creado por los autores.....	41
Figura 11. Diagrama de clases.....	41
Figura 12. Diagrama de paquetes.	42
Figura 13. Diagrama de despliegue.	42
Figura 14. Interfaz gráfica para ingreso del usuario.....	43
Figura 15. Interfaz gráfica de usuario.....	43
Figura 16. Interfaz gráfica de usuario.....	44
Figura 17. Ingreso de seguimiento del syllabus.	44
Figura 18. Envío información de temas visto según programación del syllabus.	45
Figura 19. Preguntas de encuesta de desempeño del syllabus.	45
Figura 20. Pantalla inicial del perfil de personal administrativo.....	46
Figura 21. Pantallazos para descargar los reportes.....	46
Figura 22. Imagen de resultados de reporte de desempeño del syllabus.....	47
Figura 23. Imagen de resultados de reporte de la encuesta sobre seguimiento del syllabus.	47
Figura 24. Imagen de resultados al exportar los syllabus.....	47
Figura 25. Imagen de resultados al exportar las sugerencias de los estudiantes.	48
Figura 26. Modelo Entidad Relación.	49
Figura 27. Página web para descargar el APK.	50
Figura 28. Pregunta uno del formulario de pruebas.	51
Figura 29. Pregunta dos del formulario de pruebas.....	51
Figura 30. Pregunta tres del formulario de pruebas.	52

Figura 31. Pregunta cuatro del formulario de pruebas.	52
Figura 32. Pregunta cinco del formulario de pruebas.....	53
Figura 33. Pregunta uno del formulario de pruebas funcionales.....	53
Figura 34. Pregunta dos del formulario de pruebas funcionales.	54
Figura 35. Pregunta tres del formulario de pruebas funcionales.	54
Figura 36 . Pregunta cuatro del formulario de pruebas funcionales.	55
Figura 37. Pregunta cinco del formulario de pruebas funcionales.	55
Figura 38. Pregunta uno del formulario de pruebas de diseño.	56
Figura 39. Pregunta dos del formulario de pruebas de diseño.	56
Figura 40. Pregunta tres del formulario de pruebas de diseño.	57
Figura 41. Pregunta cuatro del formulario de pruebas de diseño.	57
Figura 42. Pregunta uno del formulario de pruebas de funcionalidad.	58
Figura 43. Pregunta dos del formulario de pruebas de diseño.	58
Figura 44. Pregunta tres del formulario de pruebas de diseño.	59
Figura 45. Pregunta cuatro del formulario de pruebas de diseño.	59

Introducción

El cumplimiento del syllabus de cada asignatura del plan de estudio es de gran importancia para la Uniagustiniana y sus estudiantes, este seguimiento se podrá realizar incorporando herramientas tecnológicas y conseguir la información de manera organizada y masiva.

La aplicación móvil permite involucrar a los estudiantes y le proporcionara los datos para ingresar los temas vistos o no; y expresar sus opiniones del desempeño del syllabus a lo largo de cada semestre. Esta recopilación de datos se realiza por medio de una interfaz amigable e intuitiva, y la recopilación de datos se almacenará en una base de datos, donde se podrá exportar la información por los interesados del programa de la Uniagustiniana.

1. Título del proyecto de software

Desarrollo de una aplicación móvil, para realizar seguimiento y desempeño de los syllabus del área disciplinar del programa de tecnología en desarrollo de software de la Universitaria Agustiniana.

2. Planeación del proyecto

2.1. Objetivos del proyecto

2.1.1. Objetivo general.

Desarrollar una aplicación móvil, para realizar seguimiento y desempeño de los syllabus del área disciplinar del programa de tecnología en desarrollo de software de la Universitaria Agustiniiana, con el fin de recopilar la información suministrada, generando controles y seguimientos por parte de los estudiantes del programa.

2.1.2. Objetivos específicos.

- Crear una aplicación móvil ágil e intuitiva para realizar seguimiento al syllabus.
- Aplicar una encuesta en cada corte, con el fin de conocer la opinión del estudiante sobre el desempeño del syllabus de las asignaturas de área disciplinar.
- Implementar una base de datos con los syllabus de las asignaturas del área disciplinar igualmente los datos básicos del estudiante para el ingreso a la aplicación.
- Generar informes del seguimiento y desempeño del control de los syllabus con el fin de visualizar los temas vistos en clase y opinión expresada por los estudiantes.

2.2. Planteamiento del problema y/o necesidad

El syllabus es una herramienta según Zalbalza (2004): “[...] orientada al desarrollo de proyectos formativos, a organizar nuestra actuación no como un conjunto de acciones imprevisibles y desconectadas entre sí, sino como la puesta en práctica de un plan bien pensado y articulado” (p. 3). Es un instrumento que permite la planificación de la materia y establecer los deberes y derechos de los estudiantes y profesores dentro del ámbito académico; en la universitaria Uniagustiniana cuenta con un sistema y proceso de creación y actualización de los syllabus según registro calificado aprobado según Resolución No 22956 14 de diciembre de 2016, Ministerio de Educación Nacional.

En la Universitaria se presentan estos documentos al inicio de cada semestre a los estudiantes inscritos, uno por cada asignatura; al primer día de clase los profesores realizan una presentación de su asignatura a cargo, explicando los objetivos, metodología y el programa semana a semana, con las unidades y temas a ver a lo largo del semestre. Adicionalmente, se presenta una motivación, importancia y pertinencia de cada asignatura con sus objetivos y el valor ético dentro de la institución agustiniana.

A lo largo de la carrera de tecnología de desarrollo de software, como estudiantes se pudo evidenciar en las diferentes asignaturas vistas a través de los diferentes semestres, que no se realizaba un seguimiento del desempeño del syllabus. De igual manera, dentro de las aulas de clases algunos profesores daban a conocer su opinión, desde un punto de vista personal y su metodología de enseñanza, manifestaban que su contenido ya no estaba actualizado o que preferían un orden diferente para enseñar y que eso permitirá un aprendizaje más fácil por parte del estudiante, y en algunos casos puntuales y en común acuerdo con los estudiantes, se realizaban algunos cambios.

A partir de esto, se hace el planteamiento de cómo realizar un seguimiento y como dar a conocer la opinión del estudiante sobre el desempeño del syllabus, según cada asignatura y con esto permitiendo vigilar su desempeño, recopilando la opinión de los estudiantes y darlas a conocer a los interesados del área interdisciplinaria, permitiendo involucrar al estudiante en estos procesos.

2.3. Alcance del proyecto

El proyecto busca realizar una aplicación móvil Android donde se realice el seguimiento del cumplimiento y desempeño de los Syllabus de la Universitaria Agustiniense en la carrera Tecnólogo en desarrollo de software, a través de una recopilación de información y envío de la misma, por medio de informes al coordinador(a) del área disciplinaria de la carrera.

El seguimiento del syllabus se realiza por medio de la recopilación de la información suministrada por los estudiantes, donde ellos indicaran los temas vistos y si se cumplió o no con el cronograma expuesto semana a semana en ellos. Así mismo, para validar el desempeño del syllabus se darán unas opiniones predefinidas, por medio de preguntas cerradas con el fin de entregar dicha información por medio de informes donde facilite el análisis de los datos recibidos por parte del coordinador(a). Adicionalmente, se deja una pregunta abierta para obtener una opinión libre por parte del estudiante sobre el desempeño del syllabus.

Este proyecto inicialmente está destinado a las asignaturas del área disciplinaria de la carrera tecnólogo en desarrollo de software.

2.4. Metodología de desarrollo de software

La estructura inicial de la propuesta se realiza a través de consulta en diferentes fuentes de información tales como expertos en el área, sitios especiales y oficiales con la temática principal del proyecto utilizando la red de redes Internet.

Para el desarrollo del proyecto se aplica la (extreme programming, XP). La metodología XP es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos, es una

metodología ágil que potencia las relaciones interpersonales en equipo de trabajo, promoviendo nuevos conocimientos en los desarrolladores, para obtener soluciones sencillas y que permitan una adaptación rápida a estos cambios. Con el cliente se busca una comunicación continua y fluida con el grupo de desarrolladores.

Algunas de sus características son: respuesta rápida a los cambios constantes, la planificación es abierta con un cronograma de actividades flexible y los requisitos del cliente y el trabajo del equipo del proyecto son los principales factores de éxito de este.

El ciclo de vida para esa metodología se divide en seis fases:

Fase de exploración: se tiene y reunión con el cliente para conocer a grandes rasgos las historias de usuario que desea el cliente, con esta información el grupo de desarrolladores conoce un poco de que se trata la aplicación y busca las mejores herramientas de desarrollo, crean un bosquejo de la arquitectura del software; con esta información se realizan pruebas creando un prototipo del producto, el tiempo que lleva esta fase lo determina el tamaño del proyecto y la experiencia de los programadores.

Fase de planeación de la entrega: se programa una nueva reunión con el cliente donde se determinan cuáles son las historias de usuario más importantes para él; el grupo de desarrolladores hace una estimación de tiempo para el desarrollo de cada una y se crea un cronograma en conformidad con el cliente. Este proceso no puede tardar más de 3 meses, es este punto los programadores definen las iteraciones a realizar la velocidad de desarrollo según la cantidad de historias de usuario que desee el cliente.

Fase de iteraciones: en este punto se define un plan de entrega según las iteraciones creadas, cada uno no debe superar las tres semanas. Las iteraciones se inician con la definición de la arquitectura de software final según las historias de usuarios acordadas con el cliente y el nivel de jerarquía de cada una, así mismo, todas las iteraciones deben ser aprobadas por el cliente y de igual manera aprobar la etapa de pruebas y que cumpla con todos los requerimientos indicados para dicha iteración. Al finalizar las iteraciones se inicia la etapa de producción.

Fase de producción: en esta parte se realizan las pruebas necesarias para que el proyecto sea entregado al cliente, como las pruebas de rendimiento; también se consulta al cliente si desea algún cambio en la versión actual del producto. Según los avances de las iteraciones se puede ajustar el cronograma establecido inicialmente. Todos los cambios acordados deben tener su respectiva documentación.

Fase de mantenimiento: al trabajar con iteraciones admite tener en funcionamiento alguna de ellas mientras se entrega todo el producto, lo cual puede requerir soporte de funcionamiento junto al cliente, lo cual puede involucrar a más personal en el equipo de trabajos o una reestructura del mismo.

Fase de muerte del proyecto: al finalizar y cumplir con todos los requerimientos en las historias de usuario definidas con el cliente y que este en acuerdo con los resultados se crea un documento del sistema final y ya no se realiza ninguna modificación en la arquitectura. Si es el caso contrario y el cliente no está de acuerdo con los resultados obtenidos y se haya excedido el presupuesto se daría por finalizado el proyecto.

En este proyecto se trabajó de la siguiente manera, personajes y roles: Han Galvis Llanes, programador y tester; Doly Rodriguez Arevalo, coach y creador de idea del proyecto.

Fase I, Exploración: Se realiza una interacción entre los creadores de la aplicación donde se escucha lo que desea de esta y con esto se crean las historias del usuario y se le asignan requerimientos, costos, tiempo de entrega y recurrencia de las reuniones.

Fase II, Planeación de la Entrega: en la planeación se tendrán en cuenta varios elementos, los cuales son los siguientes:

Historias de usuario: Se realiza una reunión entre el cliente y el programador para determinar los requerimientos de la aplicación y proporciona una idea clara de lo que será el proyecto, de esta reunión se generan las siguientes historias de usuario:

Tabla 1.

Historia de usuario 1

Numero:	1
Nombre:	Presentación syllabus
Usuario:	Doly Rodriguez
Tiempo Estimado:	8 días
Descripción:	Para cada asignatura existe un syllabus donde se indican los temas a ver y la semana en que se debe de ver, este documento es presentado por el profesor el primer día de clase.

Nota: Historia de usuario 1. Elaboración propia.

Tabla 2.

Historia de usuario 2

Numero:	2
Nombre:	Seguimiento syllabus
Usuario:	Doly Rodriguez
Tiempo Estimado:	8 días
Descripción:	El estudiante debe marcar como visto o no cada tema tratado en clase y si se vio la semana asignada en el syllabus.

Nota: Historia de usuario 2. Elaboración propia.

Tabla 3.

Historia de usuario 3

Numero:	3
Nombre:	Desempeño del syllabus
Usuario:	Doly Rodriguez
Tiempo Estimado:	8 días
Descripción:	El estudiante debe indicar como le parece el syllabus existente para cada asignatura e indicar sus observaciones personales.

Nota: Historia de usuario 3. Elaboración propia.

Tabla 4.

Historia de usuario 4

Numero:	4
Nombre:	Registro estudiantes
Usuario:	Doly Rodriguez
Tiempo Estimado:	8 días
Descripción:	Se entrega la información de los estudiantes y las asignaturas inscritas para que no tenga que realizar ninguna configuración, se le entregara un usuario y contraseña.

Nota: Historia de usuario 4. Elaboración propia.

Tabla 5.

Historia de usuario 5

Numero:	5
Nombre:	Informes
Usuario:	Doly Rodriguez
Tiempo Estimado:	8 días
Descripción:	Se necesita poder exportar la información de cada información cargada por el estudiante y poder analizarla.

Nota: Historia de usuario 5. Elaboración propia.

Velocidad del proyecto: Para cada iteración se asignó una semana de tiempo para finalizar cada una.

Se entrega el siguiente cronograma:

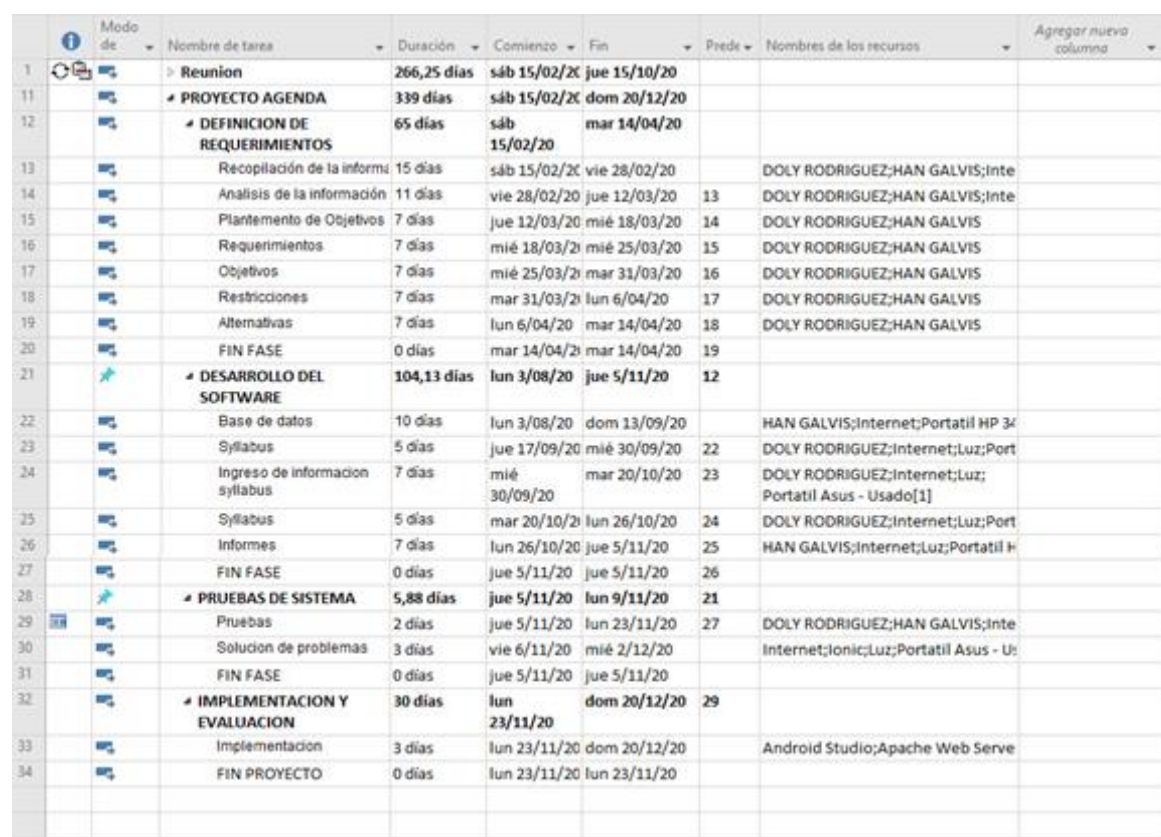


Figura 1. Diagrama de Gantt del proyecto. Elaboración propia.

Diseño: Entre los elementos más importantes que menciona XP referentes al diseño están:

Simplicidad: se diseña una aplicación ágil, intuitiva y visualmente atractiva al usuario.

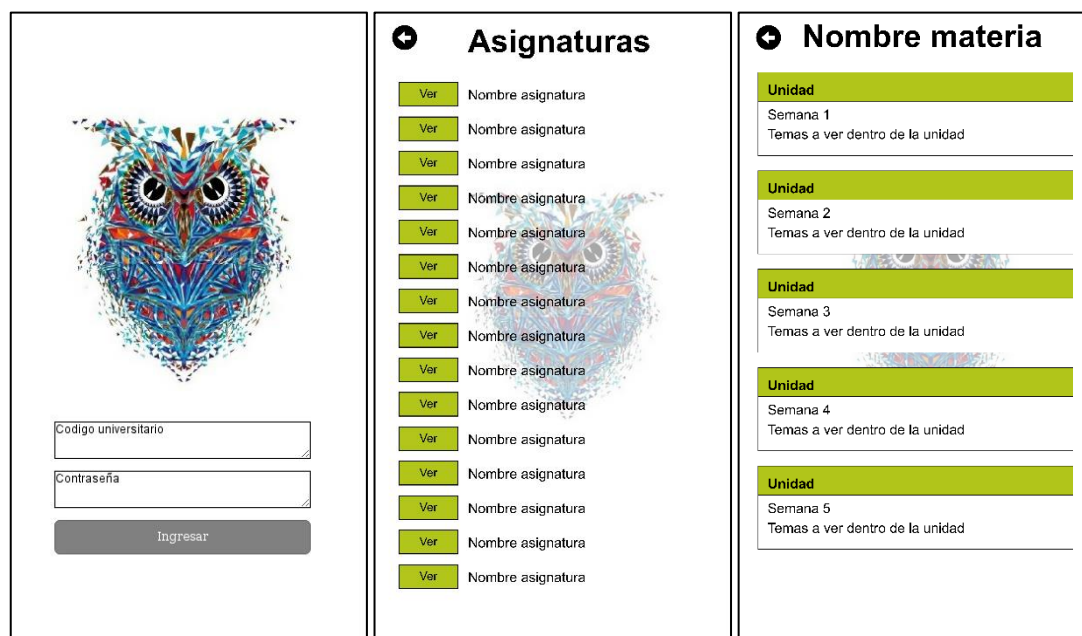


Figura 2. Mockup de inicio de sesión, visualización de asignaturas y syllabus de ellas. Elaboración propia.

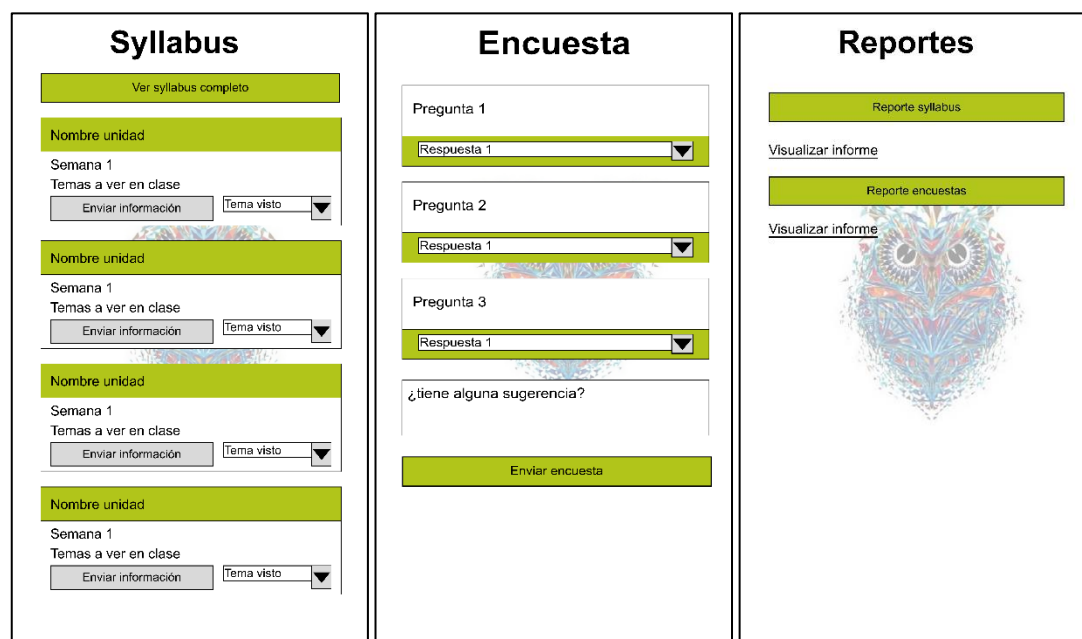


Figura 3. Mockup de seguimiento y desempeño del syllabus y exportes. Elaboración propia.

Las tarjetas CRC (clase, responsabilidad, colaboración): Las tarjetas CRC construyen para que todo el equipo de trabajo contribuya en el diseño de la aplicación, con esto busca un trabajo fundamentado en objetos y no procedimientos.

Permite identificar de clases y asociaciones que advierten del diseño del sistema y con los integrantes del equipo de desarrollo van planteando seudónimos de clases y se escriben las sugerencias.

Tabla 6.

Tarjeta CRC de Registro del estudiante

Nombre de la clase:	registroEstudiante()
Responsabilidades	Colaboradores
registrarEstudiante()	
validacionIngreso()	estudiantes()

Nota: Tarjeta CRC de Registro del estudiante. Elaboración propia.

Tabla 7.

Tarjeta CRC de gestión del syllabus

Nombre de la clase:	gestionarSyllabus()
Responsabilidades	Colaboradores
registrarSyllabus()	syllabus()
verSyllabus()	syllabus ()
seguimietoSyllabus()	
cambioStatus()	asignaturasAlumnos()
ocultarSyllabusMarcado()	syllabus()

Nota: Tarjeta CRC de Registro del syllabus. Elaboración propia.

Tabla 8.

Tarjeta CRC de la encuesta

Nombre de la clase:	encuestaSyllabus()
Responsabilidades	Colaboradores
registrarEncuesta()	encuesta()

verEncuesta()	encuesta()
cumplimientoSyllabus()	
cambioStatus()	encuesta ()

Nota: Tarjeta CRC de Registro de la encuesta. Elaboración propia.

Tabla 9.

Tarjeta CRC de informes

Nombre de la clase:	informesSyllabus()
Responsabilidades	Colaboradores
registrarAdminsitativos()	administrativos()
exportarSyllabus()	encuesta()asignaturasAlumnos()

Nota: Tarjeta CRC de Registro de informes. Elaboración propia.

Fase III, Iteraciones: el proyecto se divide en cuatro iteraciones, para un total de 4 entregas; la primera iteración se creó la base de datos de los estudiantes y los syllabus, se inicia con una sola materia y según se tengan avance se agregan las demás asignaturas; con la estructura se define la organización de la aplicación y como se mostraría la información al usuario final; se fueron realizando algunas modificaciones en las claves primarias y secundarias, y algunos cambios en la estructura de las tablas, nombre de columnas y tipo de datos.

La segunda iteración la visualización y seguimiento del syllabus, se muestra al estudiante la asignatura con su respectiva unidad y temas a ver según la semana; se realizaron medicaciones en la estructura de visualización de la información pensada inicialmente para comodidad del usuario final.

La tercera iteración la parte de desempeño del syllabus, se crea una encuesta por cada corte académico, donde se obtendrá la opinión del estudiante sobre los temas de los syllabus vistos; se crear las preguntas y respuesta para de la encuesta, se define las fechas de los cortes según programación del calendario de la universitaria Agustiniana.

Por último, la cuarta iteración es el exporte de informes, parte diseñada para los administrativos donde se muestra la información obtenida de los estudiantes; se utilizó una librería para realizar los exportes en Excel, estos se fueron ajustando para hasta obtener la información que se desea mostrar en dichos informes.

Entregas Pequeñas: cada iteración cuenta con una duración de 8 días, se determina una reunión para el final de cada iteración, en este mismo tiempo se fijaron las entregas funcionales. Esto representó una comunicación continua con el cliente.

Plan de entregas: se realiza una reunión donde se determinan la disponibilidad para la creación del proyecto y se determina que se puede dedicar 3 horas al día, según esto se reajustan las fechas de entrega de cada iteración.

La refactorización: según avanzaba el desarrollo de la aplicación, se revisó continuamente el diseño encontrando escenarios que no estuvieron contemplado al inicio del proyecto en el diseño general. Para solucionar estos inconvenientes se usó la refactorización de las partes a modificar, se buscando soluciones que afectaran lo más menos posible el código existente, algún fueron grandes cambios que no afectaron la funcionabilidad de la aplicación.

Soluciones puntuales: se utiliza HttpClientModule, una librería diseñada para Ionic, con esta librería se crea un API (application programming interface, en español interfaz de programación de aplicaciones) para la conexión a la base de datos, esto se utiliza para los métodos POST, GET de la aplicación para obtener y enviar información.

Fase IV, producción: Entre los elementos más importantes que menciona XP referentes a la codificación están:

Cliente siempre presente: a través de reuniones virtuales nos reunimos los creadores de la idea de la aplicación para solución de dudas diseño y resultados finales según la idea planteada para esta, por este mismo medio, se realizaban muestras de la aplicación para así mejorar el entendimiento de todas las partes.

Estándares y propiedad colectiva del código: para lograr una propiedad colectiva, se establece colocar comentarios en todos los procesos del código, indicando que proceso realiza y como lo realiza, así mismo, la herramienta usada para el desarrollo de la aplicación es conocida y manejada por el equipo de trabajo.

Toda la producción de código debe ser hecha en parejas: en la elaboración del proyecto no fue posible el trabajo en parejas, sin embargo, trabaja una persona directamente en el código y la otra persona a través de apoyo virtual para temas de diseño y un código más organizado.

Codificar primero la prueba: Se realizan pruebas con aplicaciones móviles ya creadas por el equipo de trabajo, esto permitió visualizar los requerimientos adicionales del código y dio claridad del resultado final deseado.

Fase V, mantenimiento: mientras se pone en funcionamiento la aplicación, se van realizando pruebas constantes, mirando el código y la estructura del requerimiento en la historia de usuario, y se va adelantando a la siguiente historia, y según se dieran los resultados, se realizaban las correcciones o se continuaba con el desarrollo de la siguiente historia. Se realizó este proceso hasta cumplir con todos los requerimientos.

Fase VI, muerte del Proyecto: Se da por concluido el proyecto al cumplir con todas las historias de cliente. Se realizan pruebas con el creador de la idea del proyecto, para confirmar su satisfacción con los resultados finales.

La aplicación será desarrollada en Ionic Framework, Ionic, (2012) "...es un kit de herramientas de interfaz de usuario móvil de código abierto para crear experiencias de aplicaciones web y nativas multiplataforma de alta calidad". Este framework nos permite exportar la aplicación para Android, un sistema operativo móvil desarrollado por Google.

La aplicación estará originalmente diseñada para utilizarse en los sistemas operativos Android, en caso de necesitar adaptarse para uno nuevo, será ajustado mediante una nueva actualización, de igual manera, se podrá realizar actualizaciones de esta para mejorar la experiencia de usuario según solicitud adicional del cliente.

3. Marco teórico y estado del arte

3.1. Marco teórico

3.1.2. Syllabus.

“Es un documento que ayuda al docente a expresar claramente a los estudiantes qué esperan aprender del curso y establece la relación entre los objetivos del curso, las metas de enseñanza aprendizaje del Departamento y la misión de la Universidad.

La palabra Syllabus proviene del latín Sillybus y está a su vez del griego que significa lista. El Syllabus esboza la lista de tópicos del curso que serán cubiertos cada semana durante el semestre, establece qué es lo que los estudiantes deben hacer para un aprendizaje efectivo y cómo será el proceso de evaluación. La principal función del Syllabus es la de actuar como un contrato entre el profesor y el estudiante, en el que cada uno se compromete con el proceso de enseñanza aprendizaje, el profesor con la información que será transmitida y cómo será transmitida y los estudiantes en lo que deberán aprender”. (Caballero, 2013, p. 1).

3.1.3. Marco institucional.

“La creación de programas de educación superior por parte de las instituciones de educación superior requiere, como primera medida, que la institución cuente con personería jurídica otorgada por la autoridad competente conforme a las normas que rigen en nuestro país y como segundo aspecto, que el programa propuesto corresponda a los propósitos de formación, según el carácter académico de la institución.

Cumplidos los presupuestos anteriores, seguidos los procesos y requisitos internos de creación del programa, la respectiva institución debe presentar ante el Ministerio de Educación Nacional la solicitud de registro calificado, que constituye el conjunto de condiciones de calidad específicas que deben demostrarse para obtener la autorización que demanda el otorgamiento del referido registro.

Actualmente, los requisitos para ofrecer y desarrollar un programa de educación superior están contenidos en la Ley 1188 de 2008 y su decreto reglamentario 1295 de 2010, conforme a estas normas, las condiciones de calidad que deben demostrarse para obtener registro calificado corresponden a condiciones institucionales y a condiciones específicas del programa”. (mineducacion, s.f., p. 1).

3.1.4. Aplicación móvil.

Una aplicación móvil es definida por el diccionario electrónico EcuRed (2018):

“una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros. Existen aplicaciones móviles gratuitas u otras de pago,

donde en promedio el 20-30% del costo de la aplicación se destina al distribuidor y el resto es para el desarrollador”. (p. 1).

3.1.5. Diseño modular.

Pes (s.f.):

“Una vez que los requisitos de un programa han sido establecidos, ya se puede iniciar la fase de diseño. En esta etapa se tiene que encontrar una solución informática al problema planteado. Dicha solución determinará cómo se va a resolver el problema. Por norma general, no suele ser fácil encontrar una solución óptima y, menos aún, cuando el problema tiene cierta envergadura. Para encontrar dicha solución, el desarrollador puede hacer uso del diseño modular. Cada uno de ellos puede tratarse de manera aislada; por tanto, la complejidad global del problema disminuirá considerablemente. Del mismo modo, si los subproblemas obtenidos siguen siendo demasiado complicados, también puede ser conveniente fragmentarlos. Y así sucesivamente, hasta llegar a subproblemas realmente sencillos”. (p. 1).

3.1.6. Base de datos.

Es definida en Oracle (s.f.):

“Una base de datos es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos”. (p. 1).

3.1.7. Android studio.

Se define según Android Studio(s.f.):

“el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes: un sistema de compilación flexible basado en Gradle, un emulador rápido y cargado de funciones, un entorno unificado donde puedes desarrollar para todos los dispositivos Android, aplicación de cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación, integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.

Variedad de marcos de trabajo y herramientas de prueba y Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros”. (p. 1).

3.1.8. Ionic.

Se define según Ionic framework(s.f.):

“es un kit de herramientas de interfaz de usuario de código abierto centrado en la creación de aplicaciones móviles de alta calidad para iOS nativo, Android nativo, y la web! Desde cero, está construido con HTML, CSS y JavaScript, por lo que los desarrolladores web deben sentirse como en casa con nuestros componentes. Nuestros componentes permiten a los desarrolladores crear experiencias nativas, todo mientras utilizan la tecnología web”. (p. 1).

3.1.9. PHP.

Se define según PHP (s.f.): “(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” (p. 1). También dice PHP (s.f.):

“Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga”. (p. 1).

3.1.10. MySQL workbench.

Se define como MySQL (s.f.): “MySQL Workbench es una herramienta gráfica para trabajar con servidores y bases de datos MySQL” (p.1). También dice MySQL (s.f.): “permite crear y administrar conexiones a servidores de bases de datos. Además de permitirle configurar parámetros de conexión, MySQL Workbench proporciona la capacidad de ejecutar consultas SQL en las conexiones de base de datos mediante el Editor de SQL integrado”. (p. 1).

3.1.11. Visual studio code.

Se define según Visual Studio Code (s.f.):

“es un editor de código optimizado con compatibilidad con operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar solo las herramientas que un desarrollador necesita para un ciclo rápido de compilación-depuración de código y deja flujos de trabajo más complejos a IDE destacados más completos”. (p. 1).

3.1.12. JavaScript.

Según definición de Eguiluz (s.f.):

“es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios". (p. 1).

3.2. Estado del arte

3.2.1. Classroom live web.

En Colombia, la empresa Classroom Live Web, ofrece una solución integral que se adapta a las necesidades de cada institución educativa integrándose a cada uno de sus procesos, ahorrando costos y ejecutando de manera innovadora y eficiente el procesamiento de la información en cada una de sus dependencias. Desarrolla páginas web para diferentes colegios en la ciudad de Bogotá, donde pueden gestionar, controlar e integrar los diferentes procesos de la institución en un solo sitio. Para unos de las instituciones educativas que tienen de clientes, crearon el módulo planeador, el cual permite realizar seguimiento de los temas vistos en clase y permite que los estudiantes pueden adjuntar evidencias de lo realizado durante las actividades en clase.

3.2.2. Sistema de seguimiento de syllabus.

En Ecuador, Lily del Carmen Flor Gavilanes en el 2014, en la Universidad Tecnológica Israel, elabora un proyecto para mejorar el proceso de seguimiento de syllabus de los docentes de la Universidad, tuvo como principal objetivo mejorar el proceso de seguimiento de syllabus de los docentes; para cumplir con dicho objetivo, se planteó desarrollar un sistema en ambiente web, obteniendo como resultados mejoramiento del proceso principal del seguimiento de syllabus, un proceso dinámico y optimizado del proceso de seguimiento de syllabus y reportes de la información ingresada en los diferentes módulos. Para la elaboración uso metodología XP, una metodología ágil centrada en obtener un ambiente amigable y cómodo entre el equipo de trabajo como factor importante para el triunfo en el proyecto, cuidando del aprendizaje del equipo de trabajo, y manteniendo un adecuado clima de trabajo.

3.2.3. Herramienta web para instrumento de seguimiento de syllabus.

En este mismo país en el 2016, el Dr. Marcelo Alejandro López Zea en la Universidad Regional Autónoma de los Andes, elabora un proyecto para el perfeccionamiento del syllabus de derecho administrativo en la carrera de derecho de Uniandes, ya que el desconocimiento del Derecho Administrativo provoca que los graduados al momento de ejercer la profesión, recién comiencen a

estudiar las normas referentes a la administración pública, provocando una inseguridad para su defendido; buscaba que a través la investigación de campo diera a conocer las necesidades y las falencias que tiene el syllabus, encontró que el silabo no cumplía con las misión propia de la asignatura, ni las expectativas de los estudiantes y de los profesionales del derecho y debían priorizar la teoría con la práctica, y que el perfeccionamiento del Syllabus, mejorará el conocimiento del estudiante del derecho.

3.2.4. Cibercolegios.

Es un Sistema de Gestión Educativa en línea que permite la integración de los procesos de comunicación, académicos y administrativos de Instituciones Educativas de preescolar, básica y media a nivel nacional. Cuentan con una robusta infraestructura tecnológica, estrictos controles de calidad y un talento humano competente y comprometido, brindando el soporte y acompañamiento que cada uno de sus usuarios.

Dentro de sus planes incluye un módulo de gestión Académica, con la cual buscan fortalecer el seguimiento al rendimiento académico de los alumnos y optimizar el control de calificaciones, reportes y actividades de la institución educativa.

3.2.5. EduColombia.

Es una fundación liderados por profesionales de diferentes disciplinas, desde el año 2005. Es un sistema integral para gestión educativa a través de una aplicación web, diseñado para la gestión académica y curricular de las instituciones de educación preescolar, básica y media, permitiendo que toda la comunidad educativa se comunique con facilidad y obtenga información actualizada en línea; la información la manejan distribuida en módulos, uno de ellos es el módulo de temas de estudio, allí se puede crear los temas a tratar durante el periodo los cuales pueden llevar imágenes, vínculos a páginas de interés y videos para que el alumno desde su casa pueda tener información suficiente para hacer las tareas que llegare a tener. También dispone de material a descargar para ampliar los conocimientos del Tema, estos se pueden evaluar y el Padre de familia vera en tiempo real la calificación.

4. Especificación de requisitos de software (IEEE 830)

4.1. Perspectiva del producto

La aplicación móvil Syllapp está diseñada con el fin de realizar un seguimiento y desempeño de los syllabus del área disciplinar del programa tecnología en desarrollo de software de la Universitaria Agustiniiana a través de teléfonos móviles, permitiendo ser accesible en todo momento de una forma ágil e intuitiva con disponibilidad de 7 días por 24 horas.

4.2. Funcionalidad del producto

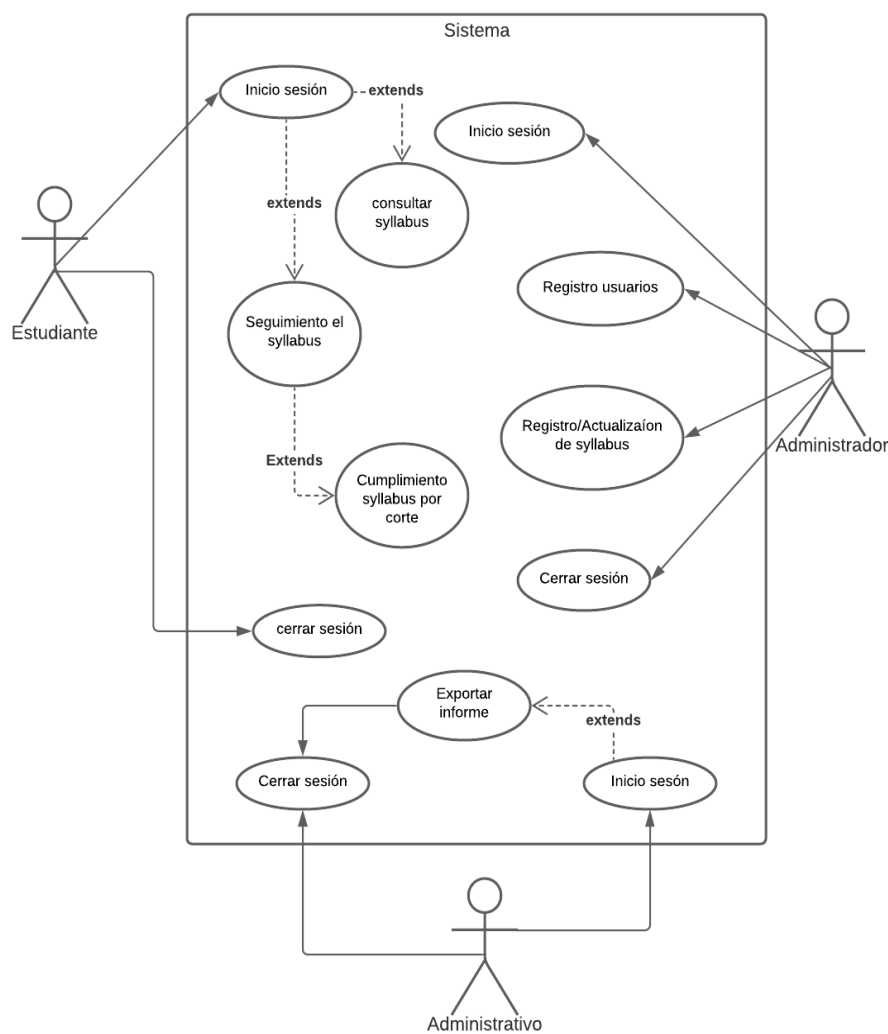


Figura 4. Funcionalidad del producto. Elaboración propia.

4.3. Características de los usuarios

Tabla 10.

Características del usuario estudiante

Tipo de usuario:	Estudiante
Formación:	Tecnología en desarrollo de software
Actividades:	Ingreso y envío de información de seguimiento y cumplimiento del syllabus.

Nota. Características del usuario estudiante. Elaboración propia.

Tabla 11.

Administrativo Uniagustiniana del área disciplinar

Tipo de usuario:	Administrativo Uniagustiniana del área disciplinar
Formación:	Profesional Uniagustiniana del área disciplinar
Actividades:	Exporta informes de información ingresada por los estudiantes

Nota. Características del usuario del personal administrativo de la Uniagustiniana del área disciplinar. Elaboración propia.

Tabla 12.

Administrador de la base datos

Tipo de usuario:	Administrador de base datos
Formación:	Tecnología en desarrollo software
Actividades:	Control y manejo de la base de datos

Nota. Características del usuario administrador de la base de datos. Elaboración propia.

4.4. Restricciones

La aplicación requiere internet para su funcionamiento.

Para el uso de la aplicación el estudiante debe existir en la base de datos.

Funciona en teléfonos móviles con sistema operativo Android.

El servidor debe ser capaz de responder las consultas de manera concurrente.

El sistema se trazará como un modelo cliente/servidor.

4.5. Suposiciones y dependencias

Al realizar cambios en el syllabus se requerirá una actualización en la base de datos de la aplicación.

Se debe ingresar la información de los estudiantes nuevos a la base de datos.

Versión del sistema operativo Android superior a 8.0.0.

4.6. Requisitos específicos

4.6.1. Actores/roles.

Tabla 13.

Rol del estudiante

Actor:	Estudiante
Descripción:	El usuario será registrado en el sistema y podrá realizar consultas, reporte de seguimiento y desempeño del syllabus
Características:	El usuario consultará las asignaturas inscrita con sus respectivos syllabus, y reporta seguimiento y desempeño del syllabus.
Relación:	El usuario de los estudiantes se relaciona con los administrativos de la Uniagustiniana del área disciplinar y administrador de base de datos.
Referencias:	Ver syllabus. Reporta seguimiento y desempeño del syllabus.

Nota. Rol del estudiante. Elaboración propia.

Tabla 14.

Rol del personal administrativo Uniagustiniana del área disciplinar

Actor:	Administrativos Uniagustiniana del área disciplinar
Descripción:	El usuario será registrado en el sistema y podrá realizar consultas.
Características:	El usuario consultará informes de seguimiento y desempeño del syllabus.
Relación:	Se relaciona con los usuarios de estudiante, quien brindara la información de seguimiento y desempeño del syllabus.
Referencias:	Generar informe.

Nota. Rol del personal administrativo Uniagustiniana del área disciplinar. Elaboración propia.

Tabla 15.

Rol del administrador de base de datos

Actor:	Administrador de base de datos
Descripción:	El usuario será registrado en el sistema y podrá realizar consultas, modificación y eliminación de datos.

Características:	El usuario ingresa, consulta y/o elimina los usuarios y sus datos correspondientes.
Relación:	Se relaciona con los usuarios de estudiante y administrativos Uniagustiniana del área disciplinar.
Referencias:	Ingreso, modificación y eliminación de datos.

Nota. Rol del administrador de base de datos. Elaboración propia.

4.6.2. Requisitos funcionales.

Tabla 16.

Requisitos funcionales de la base de datos

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Base de datos
Características:	Almacenar la información sobre los estudiantes y los syllabus
Descripción del requerimiento:	Mostrará información general acerca de las asignaturas del área disciplinar y los datos del estudiante.
Requerimiento NO funcional:	RNF01 RNF03
Prioridad del requerimiento:	Alta

Nota. Requisitos funcionales de la base de datos. Elaboración propia.

Tabla 17.

Requisitos funcionales del seguimiento syllabus

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Seguimiento syllabus
Características:	El sistema visualizará los temas por signatura según el syllabus a ver cada semana.
Descripción del requerimiento:	El estudiante podrá marca los temas vistos en clase según el syllabus
Requerimiento NO funcional:	RNF01 RNF03

Prioridad del requerimiento:	Alta
------------------------------	------

Nota. Requisitos funcionales del seguimiento syllabus. Elaboración propia.

Tabla 18.

Requisitos funcionales del desempeño syllabus

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Desempeño del syllabus
Características:	El sistema mostrara una encuesta en cada corte, para conocer la opinión del estudiante.
Descripción del requerimiento:	El estudiante podrá marca los temas vistos en clase según el syllabus
Requerimiento NO funcional:	RNF01 RNF03
Prioridad del requerimiento:	Alta

Nota. Requisitos funcionales del desempeño syllabus. Elaboración propia.

Tabla 19.

Requisitos funcionales de los informes sobre syllabus

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Informes
Características:	Garantizar el poder dar información al usuario
Descripción del requerimiento:	Generar reportes por módulo del desempeño para que el usuario pueda tener accesos a los datos generales.
Requerimiento NO funcional:	RNF01 RNF03
Prioridad del requerimiento:	Alta

Nota. Requisitos funcionales de los informes sobre syllabus. Elaboración propia.

4.6.3. Diagrama de casos de uso.

Diagrama general:

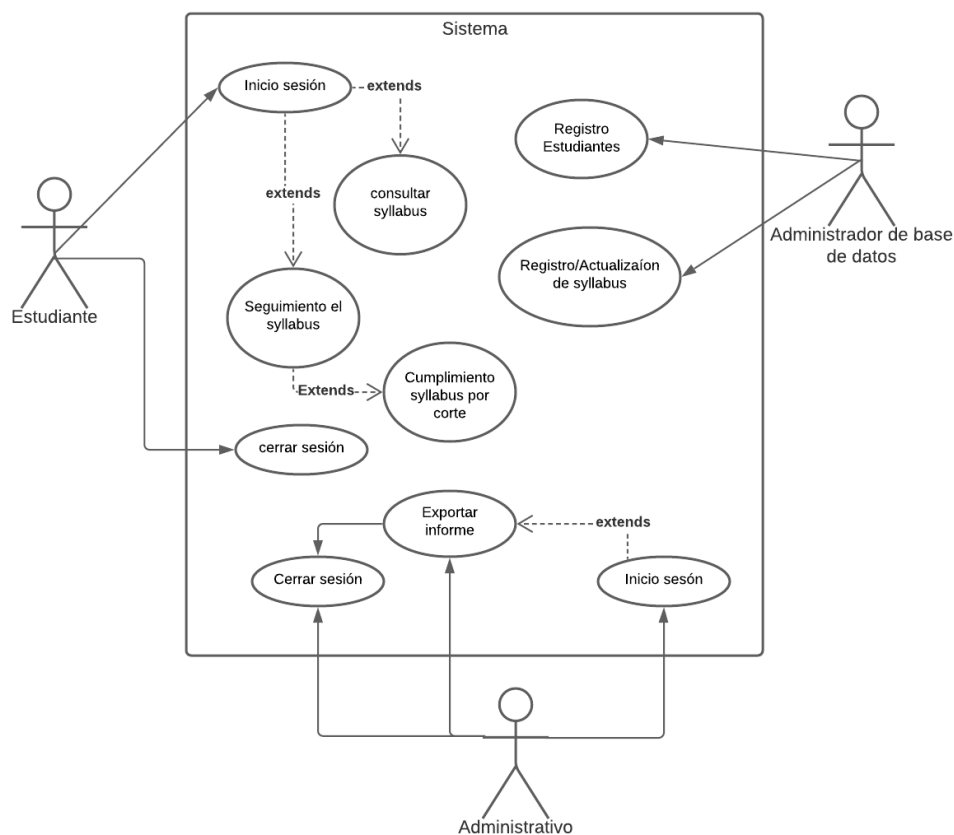


Figura 5. Caso de uso general creado por los autores. Elaboración propia.

A continuación, los casos de uso específicos:

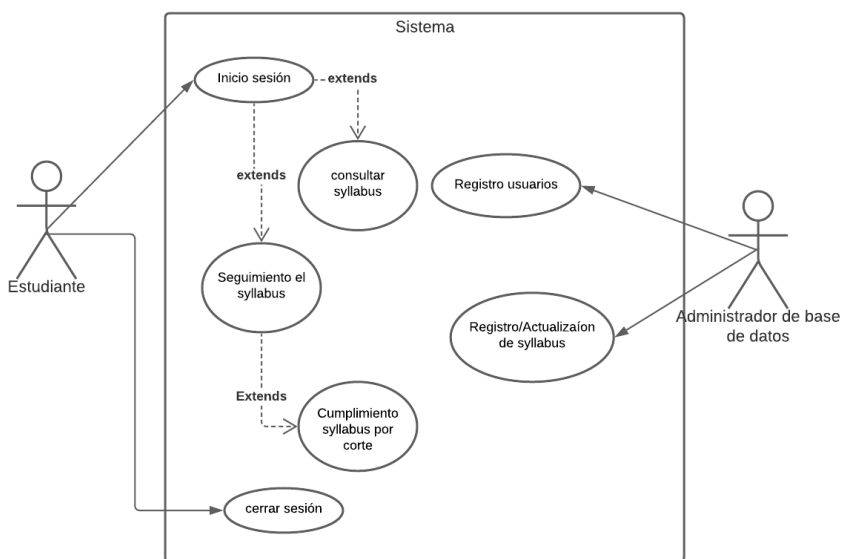


Figura 6. Caso de uso específico: estudiante. Elaboración propia.

Administrativos Uniagustiniana del área disciplinar:

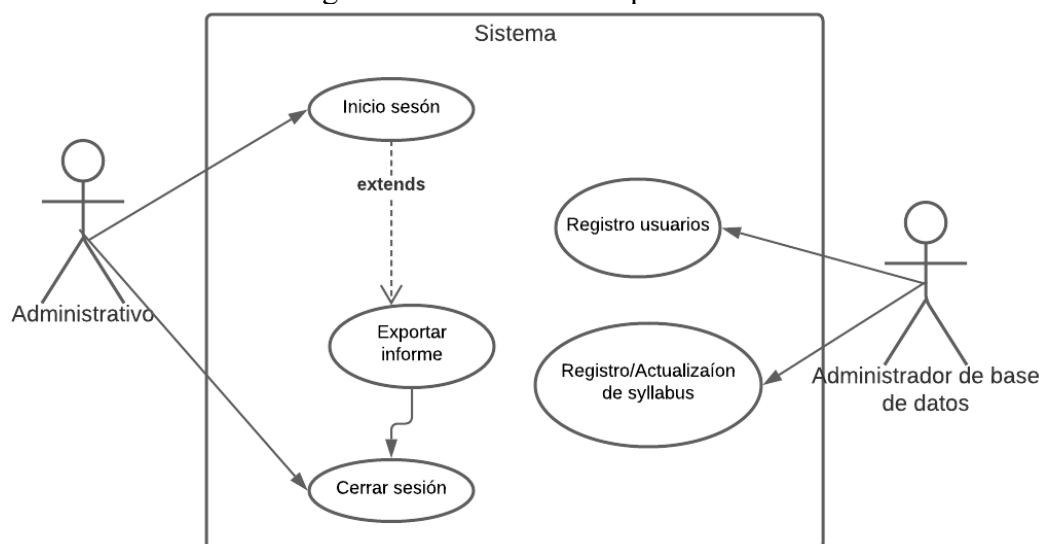


Figura 7. Caso de uso específico: Administrativos Uniagustiniana del área disciplinar. Elaboración propia.

4.6.4. Especificaciones de los casos de uso.

Tabla 20.

Caso de uso del estudiante

Caso de Uso	Estudiante
Actores	Estudiantes
Tipo	Primordial
Referencias	Ingreso seguimiento y desempeño del syllabus
Precondición	el estudiante y syllabus deben estar registrados en el sistema
Postcondición	Visualizar la información ingresada por el estudiante
Descripción	El estudiante ingresa el seguimiento y desempeño en el sistema.
Resumen	El estudiante ingresa en el sistema con su usuario correspondiente, ingresa a realizar el seguimiento y desempeño en el sistema.

Nota. Caso de uso del estudiante. Elaboración propia.

Tabla 21.

Caso de uso del administrativo de la Uniagustiniana

Caso de Uso	Administrativos Uniagustiniana del área disciplinar
Actores	Administrativo
Tipo	Primordial
Referencias	Exporte de informes de seguimiento y desempeño del syllabus
Precondición	El administrativo puede exportar los datos registrados en el sistema

Postcondición	Visualizar la información ingresada por el estudiante
Descripción	El administrativo puede visualizar y exportar los datos registrados en el sistema
Resumen	El administrativo ingresa en el sistema con su usuario correspondiente, ingresa puede visualizar y exportar los datos registrados en el sistema del seguimiento y desempeño en el sistema.

Nota. Caso de uso del administrativo de la Uniagustiniana. Elaboración propia.

4.7. Requisitos de rendimiento

Se contará con un servidor de base de datos en línea, donde los usuarios podrán interactuar simultáneamente con el servidor; se estima que el sistema podrá soportar unos 15 datos de envío en simultánea.

4.8. Restricciones de diseño

La aplicación está diseñada solo para dispositivos móviles con sistema operativo Android.

4.9. Atributos del software del sistema

4.9.1. Requisitos no funcionales.

Tabla 22.

Requisitos no funcionales del interfaz del sistema

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema
Características:	El sistema mostrara una interfaz sencilla de fácil uso y entendimiento a los usuarios del sistema.
Descripción del requerimiento:	El sistema cuenta con una interfaz de uso intuitivo y sencilla.
Prioridad del requerimiento:	Alta

Nota. Requisitos no funcionales del interfaz del sistema. Elaboración propia.

Tabla 23.

Requisitos no funcionales de ayuda en el uso del sistema

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Ayuda en el uso del sistema
Características:	La interfaz del usuario deberá de presentar un sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema.

Descripción del requerimiento:	La interfaz debe contar con un sistema de ayuda permitiendo al usuario un buen uso de aplicación, de igual manera para el administrador de la aplicación.
Prioridad del requerimiento:	Alta

Nota. Requisitos no funcionales de ayuda en el uso del sistema. Elaboración propia.

Tabla 24.

Requisitos no funcionales de mantenimiento

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Mantenimiento
Características:	El sistema se le podrá realizar mantenimiento por medio de actualizaciones según cambios del contenido de los syllabus del área disciplinar.
Descripción del requerimiento:	El sistema cuenta con una documentación de fácil entendimiento para realizar actualización en la base de datos o aplicación permitiendo realizar mantenimiento de manera intuitiva.
Prioridad del requerimiento:	Alta

Nota. Requisitos no funcionales de mantenimiento. Elaboración propia.

Tabla 25.

Requisitos no funcionales de seguridad en información

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Seguridad en información
Características:	El sistema garantizara a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Descripción del requerimiento:	Se garantiza la seguridad del sistema y la información de los datos que reposan en ella, como usuarios y contraseñas.
Prioridad del requerimiento:	Alta

Nota. Requisitos no funcionales de seguridad en información. Elaboración propia.

4.9.2. Portabilidad.

La aplicación será implementada en teléfonos móviles con sistema operativo Android en versiones superiores a 8.0.0.

4.9.3. Disponibilidad.

La aplicación tendrá una disponibilidad continua para los usuarios de 7 días por 24 horas.

4.9.4. Mantenibilidad.

La aplicación podrá ser modificada según necesidad de cambios correctivo o por actualización de los syllabus, esto se podrá realizar por módulos con un bajo impacto en los demás entornos, esto podrá realizarse a través de actualizaciones de la aplicación.

5. Diseño del software (ISO -12207-1)

5.1. Diseño de la arquitectura de software

Para el diseño de la arquitectura de la aplicación móvil se trabajó con el modelo C4, Simón Brown es el creador del modelo de arquitectura de software C4, que es un enfoque simple para crear mapas de su código.

5.1.1. El diagrama de contexto del sistema.

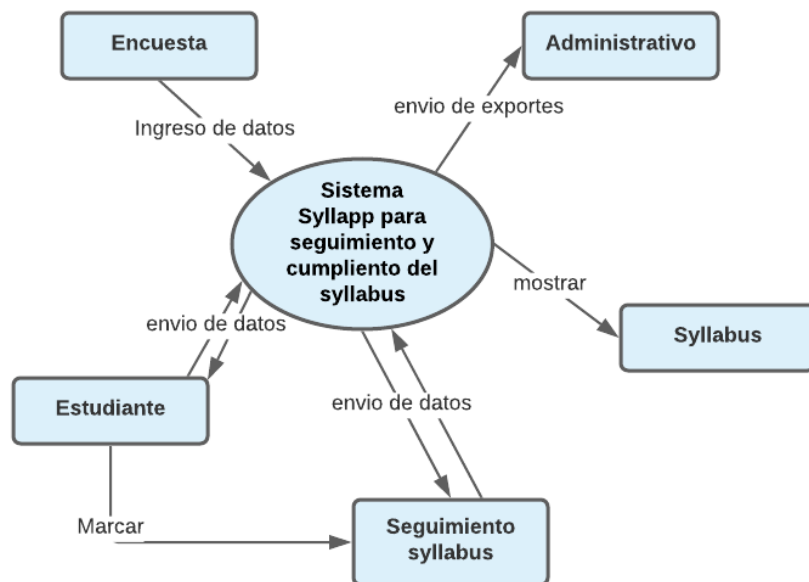


Figura 8. Diagrama de contexto del sistema. Elaboración propia.

5.1.2. El diagrama del contenedor.

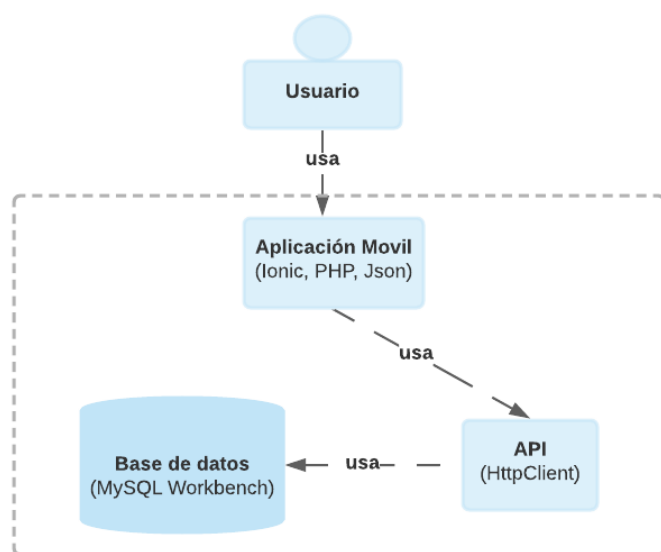


Figura 9. Diagrama del contenedor. Elaboración propia.

5.1.3. El diagrama de componentes.

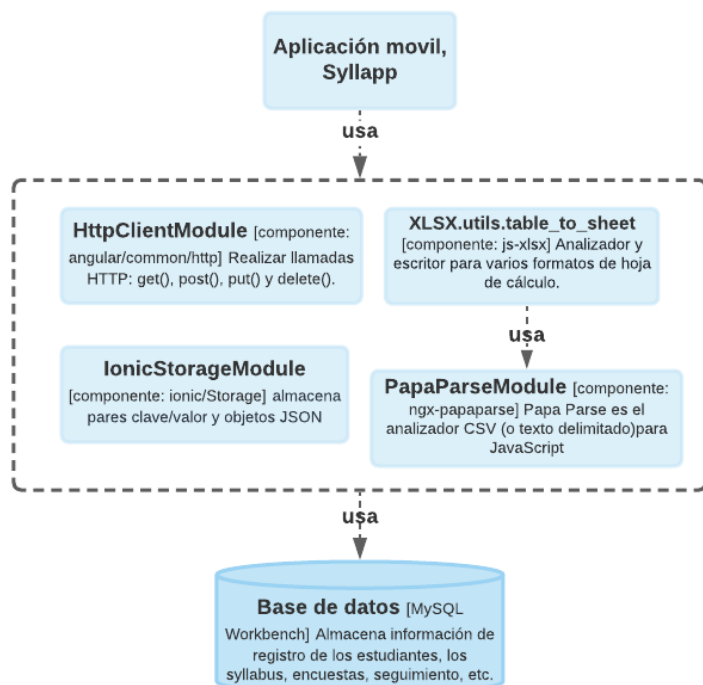


Figura 10. Diagrama de componentes, creado por los autores. Elaboración propia.

5.2. Diseño detallado del software

5.2.1. Diagrama de clases.

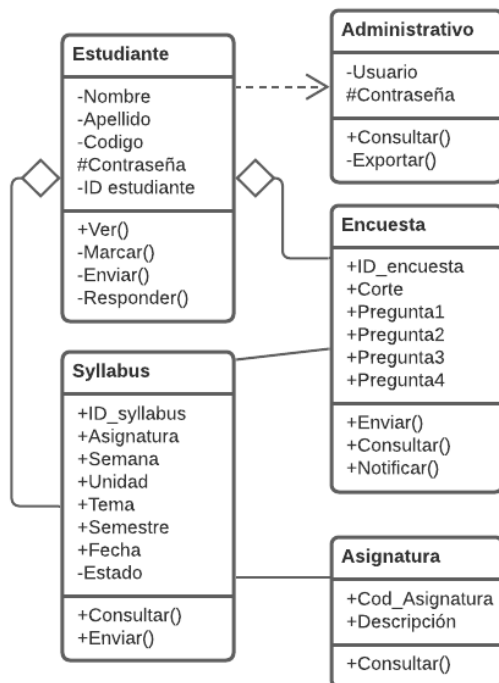


Figura 11. Diagrama de clases. Elaboración propia.

5.2.2. Diagrama de paquetes.

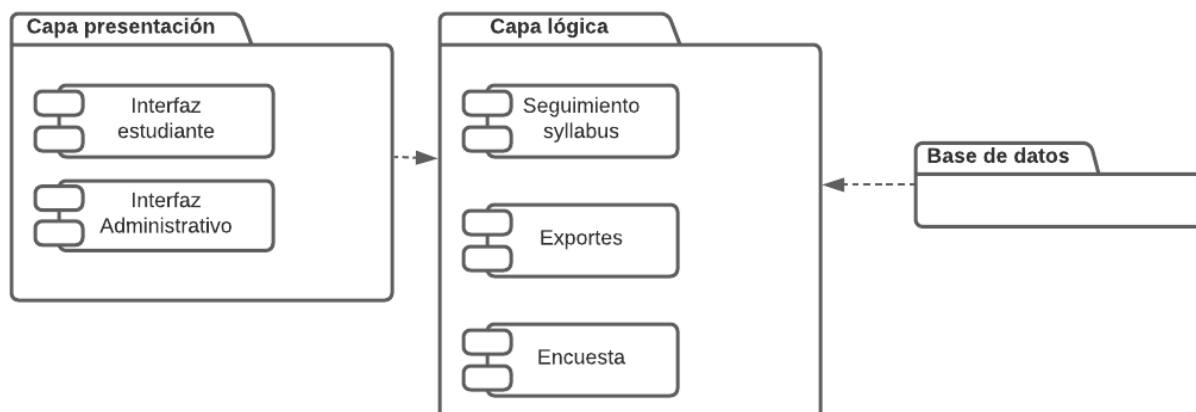


Figura 12. Diagrama de paquetes. Elaboración propia.

5.2.3. Diagramas de despliegue.

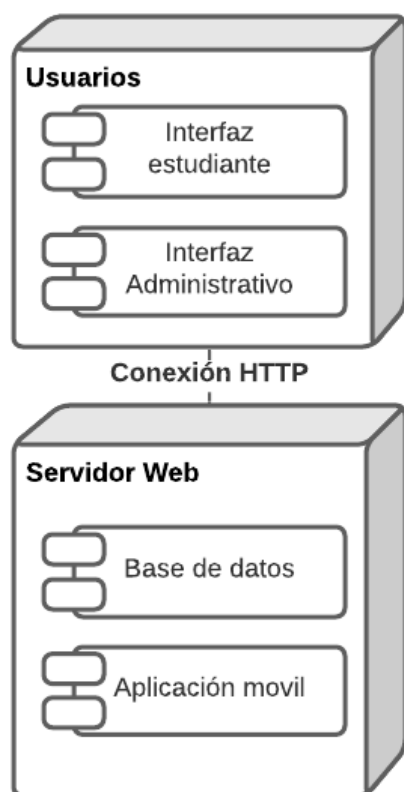


Figura 13. Diagrama de despliegue. Elaboración propia.

5.3. Diseño de la interfaz

5.3.1. Interfaz gráfica de usuario.



Figura 14. Interfaz gráfica para ingreso del usuario. Elaboración propia.

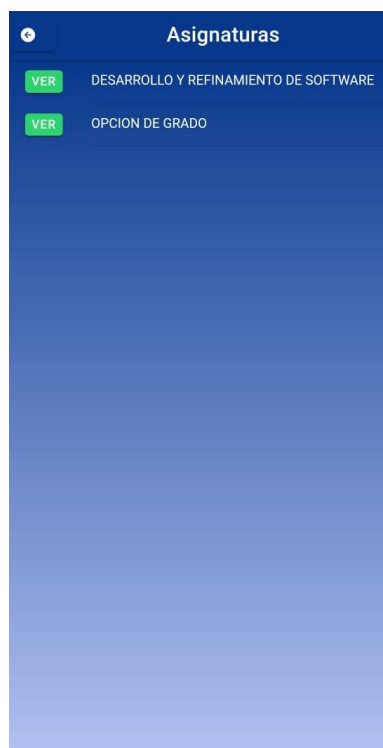


Figura 15. Interfaz gráfica de usuario. Elaboración propia.



Figura 16. Interfaz gráfica de usuario. Elaboración propia.

5.3.2. Interfaces de entrada.



Figura 17. Ingreso de seguimiento del syllabus. Elaboración propia.

The screenshot shows a mobile application interface with a dark blue header. At the top, it says "Bienvenido 2220181002" and "VER SYLLABUS COMPLETO". Below this is a dropdown menu with the text "¿Tema visto segun programación del ...". A green button labeled "ENVIAR" is positioned below the dropdown. A modal dialog box is overlaid on the screen, containing the question "¿Tema visto segun programación del syllabus?" and two radio button options: "SI" and "NO". At the bottom of the modal are "CANCEL" and "OK" buttons. The background shows a list of syllabus topics, including "10. DESARROLLO Y REFINAMIENTO DE SOFTWARE" and "11. DESARROLLO Y REFINAMIENTO DE SOFTWARE".

Figura 18. Envió información de temas visto según programación del syllabus. Elaboración propia.

The screenshot shows a mobile application interface with a dark blue header. At the top, it says "Encuesta Corte uno" and "ENVIAR ENCUESTA". Below this are four survey questions, each followed by a green button for response:

- ¿La presentación del syllabus es adecuada y ordenada?
- ¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?
- ¿El trabajo realizado en el aula permite el cumplimiento de los resultados de aprendizaje que constan en el syllabus?
- ¿Tiene alguna sugerencia?

At the bottom of the form is a large green button labeled "ENVIAR ENCUESTA".

Figura 19. Preguntas de encuesta de desempeño del syllabus. Elaboración propia.

5.3.3. Interfaces de salida.



Figura 20. Pantalla inicial del perfil de personal administrativo. Elaboración Propia.

Reporte Syllabus				Reporte Encuesta			Reporte seguimiento Syllabus				Reporte Sugerencias		
DESCARGAR EXCEL				DESCARGAR EXCEL			DESCARGAR EXCEL				DESCARGAR EXCEL		
Asignatura	Semana	unidad	Tema	preguntas	Respuestas	Cantidad	Cod. Estudiante	Asignatura	unidad	Te	Corte	Sugerencias	
DESARROLLO Y REFINAMIENTO DE SOFTWARE	1	Presentación del programa	Presentación del programa y contextualizada con la actualidad e importancia del refinamiento de software	¿La presentación del syllabus es adecuada y ordenada?		1	2220181034	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	SI	1	esta es una prueba	
				¿La presentación del syllabus es adecuada y ordenada?	No	6	2220181002	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	SI	1	prueba desde usuario	
DESARROLLO Y REFINAMIENTO DE SOFTWARE	2	Construir los primeros niveles de diseño básico y detallado del sistema	"Definir la arquitectura implementada del modelo abstrayendo lógica Funcional"	¿La presentación del syllabus es adecuada y ordenada?	No Sé	34	2220181003	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	NI	1	vv	
				¿La presentación del syllabus es adecuada y ordenada?	Si	15	2220181021	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	SI	1	Ninguna	
DESARROLLO Y REFINAMIENTO DE SOFTWARE	3	Iniciar la implementación de los requerimientos en busca de la solución del problema planteado	"La creación de la base de datos. La creación de los esquemas. Construir e implementar arquitectura propuesta"	¿La presentación del syllabus es adecuada y ordenada?		3	2220181006	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	SI	1	2	
				¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No	15	2220181006	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	SI	1	sadsad	
DESARROLLO Y REFINAMIENTO DE SOFTWARE	4	Definir e identificar las fases bajo el modelo unificado	Introducir el modelo RUP proyecto propuesto	¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No	15	2220181034	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	SI	1	hvh	
				¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No Sé	10	2220181002	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	SI	1	hvxj	
DESARROLLO Y REFINAMIENTO DE SOFTWARE	5	Parcial I	Parcial I	¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No Sé	10	2220181003	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	SI	1	Ninguna	
				¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No Sé	10	2220181003	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	SI	1	g	

Figura 21. Pantallazos para descargar los reportes. Elaboración Propia.

	A	B	C	D
1	Cod.Estudiante	Asignatura	unidad	Tema visto
2	2220181002	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Presentación del programa	SI
3	2220181002	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Construir los primeros niveles de diseño básico y detallado del sistema	NO
4	2220181002	DESARROLLO Y REFINAMIENTO DE SOFTWARE	Iniciar la implementación de los requerimiento en busca de la solución del problema planteado	SI
5				
6				

Figura 22. Imagen de resultados de reporte de desempeño del syllabus. Elaboración propia.

	A	B	C	D
1	preguntas	Respuestas	Cantidad	
2	¿La presentación del syllabus es adecuada y ordenada?	No	4	
3	¿La presentación del syllabus es adecuada y ordenada?	No Sé	7	
4	¿La presentación del syllabus es adecuada y ordenada?	Si	11	
5	¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No	6	
6	¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	No Sé	10	
7	¿Los resultados de aprendizaje propuestos en el syllabus fueron alcanzados al final de asignatura?	Si	6	
8	¿El trabajo realizado en el aula permite el cumplimiento de los resultados de aprendizaje que constan en el syllabus?	No	11	
9	¿El trabajo realizado en el aula permite el cumplimiento de los resultados de aprendizaje que constan en el syllabus?	No Sé	5	
10	¿El trabajo realizado en el aula permite el cumplimiento de los resultados de aprendizaje que constan en el syllabus?	Si	6	
11				
12				

Figura 23. Imagen de resultados de reporte de la encuesta sobre seguimiento del syllabus.

Elaboración propia.

	A	B	C	D
1	Asignatura	Semana	unidad	Tema
2	DESARROLLO Y REFINAMIENTO DE SOFTWARE	1	Presentación del programa	Presentación del programa y contextualiza
3	DESARROLLO Y REFINAMIENTO DE SOFTWARE	2	Construir los primeros niveles de diseño básico y de	Definir la arquitectura, implementando de
4	DESARROLLO Y REFINAMIENTO DE SOFTWARE	3	Iniciar la implementación de los requerimiento en bi	La creación de la base de datos La creaci
5	DESARROLLO Y REFINAMIENTO DE SOFTWARE	4	Definir e identificar las fases bajo el modelo unifica	Introducir el modelo RUP al proyecto prop
6	DESARROLLO Y REFINAMIENTO DE SOFTWARE	5	Parcial I	Parcial I
7	DESARROLLO Y REFINAMIENTO DE SOFTWARE	6	Determinar las fases de inicio y sus iteraciones	Implementar la fase de inicio y sus artefac
8	DESARROLLO Y REFINAMIENTO DE SOFTWARE	7	Presentar los artefactos adecuados y refinados	Implementar y adecuar los artefactos a la
9	DESARROLLO Y REFINAMIENTO DE SOFTWARE	8	Definir los requerimientos refinados y desarrollar ar	Presentar los productos entregables y con
10	DESARROLLO Y REFINAMIENTO DE SOFTWARE	9	Definir los requerimientos y observar las iteraciones	Implementar el diseño básico y detallado :
11	DESARROLLO Y REFINAMIENTO DE SOFTWARE	10	Aplicar pruebas de retroalimentación a los artefacto	Implementar esta clasificación y revisar el
12	DESARROLLO Y REFINAMIENTO DE SOFTWARE	11	Parcial II	Parcial II
13	DESARROLLO Y REFINAMIENTO DE SOFTWARE	12	Refinamiento a partir de la granularidad	Refinamiento a partir de tipos de comport
14	DESARROLLO Y REFINAMIENTO DE SOFTWARE	12	Dotar al sistema de toda la implementación propue	Implementar la arquitectura a partir de la r
15	DESARROLLO Y REFINAMIENTO DE SOFTWARE	14	Refinar el diseño y realizar las transiciones según lc	Implementar el modelo según el diagrama
16	DESARROLLO Y REFINAMIENTO DE SOFTWARE	15	Entrega del proyecto	Implementación del proyecto
17	DESARROLLO Y REFINAMIENTO DE SOFTWARE	16	Evaluacion.Final	Evaluacion.Final

Figura 24. Imagen de resultados al exportar los syllabus. Elaboración propia.

	A	B	C	D	E	F	G	H
1	Corte	Sugerencias						
2		1 esta es una prueba						
3		1 prueba desde usuario						
4		1 Ninguna						
5		1 Ninguna						
6		1 Ninguno						
7		1 Ninguna						
8		1 no						
9		1 no						

reportes_seguimiento

Figura 25. Imagen de resultados al exportar las sugerencias de los estudiantes. Elaboración propia.

6. Implementación

6.1. Plataformas de desarrollo

La aplicación es desarrollada en Ionic Framework, PHP, JavaScript y Angular; para la base de datos se usó MySQL Workbench, como editor de código se maneja visual Studio Code y Android Studio para la exportación del paquete de instalación de la aplicación.

6.2. Base de datos

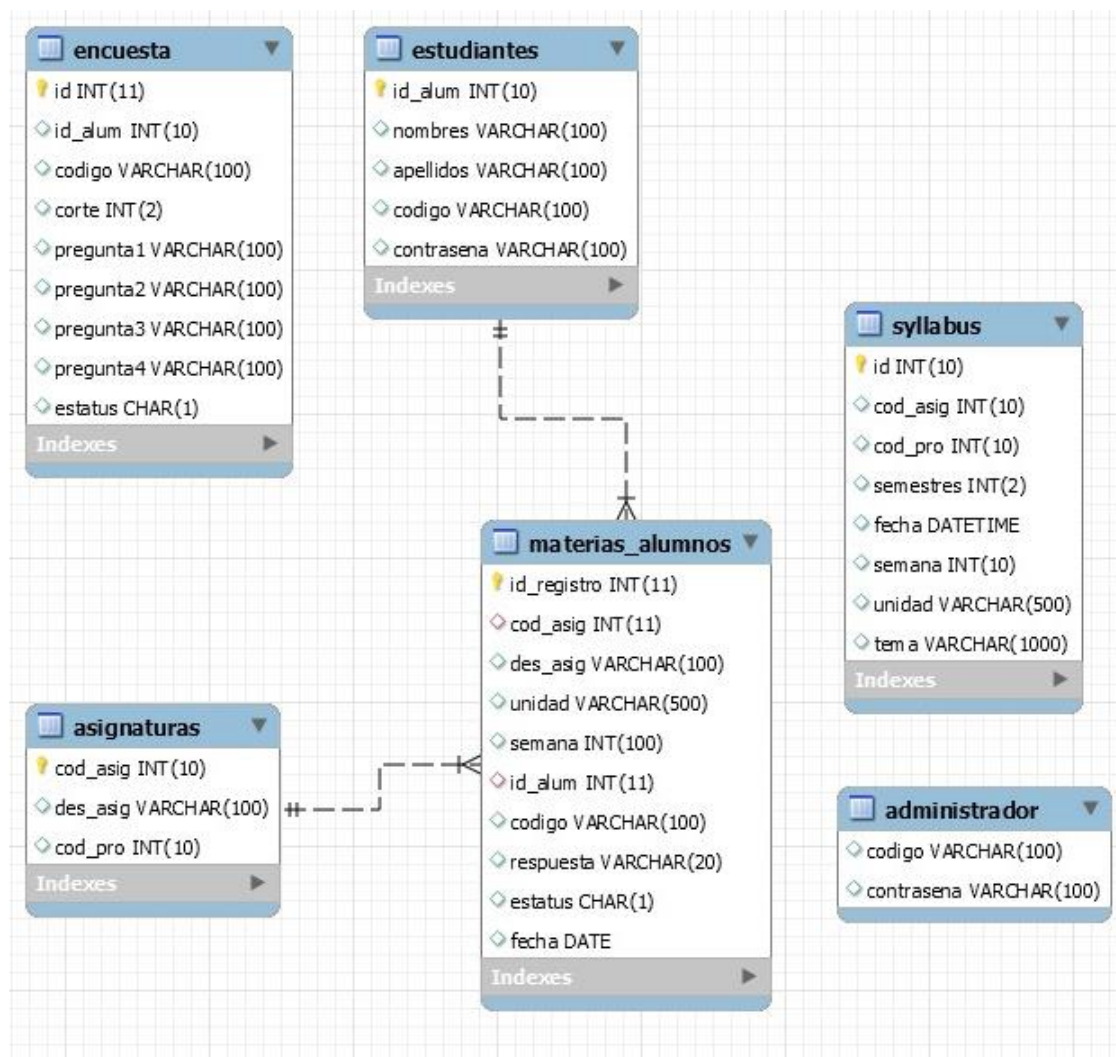


Figura 26. Modelo Entidad Relación. Elaboración propia.

6.3. Infraestructura de hardware y redes

La aplicación móvil es desarrollada para dispositivos móviles con sistema operativo Android con versiones superiores a 8.0.0. sin importar la marca del dispositivo; para el uso de la aplicación se requiere conexión a internet.

Para el ingreso a la aplicación el estudiante debe estar registrado con anterioridad en la base de datos y el administrador de la misma debe suministrarle el usuario y contraseña, quien es el encargado de ingresar, modificar o eliminar la información en la base de datos.

Para acceder a la aplicación se debe ingresar a la siguientes página web <https://practicauniagustinianasyllabus.000webhostapp.com/> , en la cual aparece una guía de uso e instrucciones de cómo realizar la descarga e instalación de la aplicación.



Figura 27. Página web para descargar el APK. Elaboración propia.

7. Pruebas del software

7.1. Pruebas del software

Con las pruebas de software se busca probar la aplicación en sus diferentes procesos y comprobar su correcto funcionamiento. Estas pruebas se realizaron con profesionales del área del software, las preguntas se realizan a través de un formulario, para así recopilar y validar las posibles mejoras que requiere la aplicación.

En el formulario realizado se utilizan preguntas con respuestas cerradas y una pregunta abierta; de dichas pruebas se obtienen los siguientes resultados:

Pruebas de sistema:

¿Cada parte de la interfaz comienza con un título que describe el contenido de la pantalla?
3 respuestas

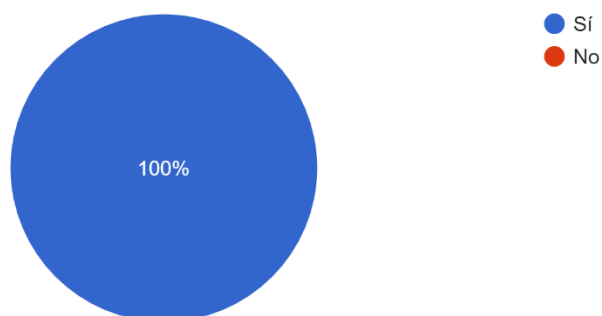


Figura 28. Pregunta uno del formulario de pruebas. Elaboración propia.

¿El esquema de diseño y estética es coherente en toda la aplicación?
3 respuestas

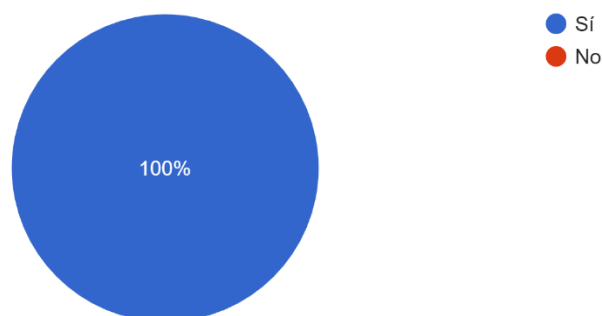


Figura 29. Pregunta dos del formulario de pruebas. Elaboración propia.

¿Los tiempos de respuesta son adecuados para cada tarea?

3 respuestas

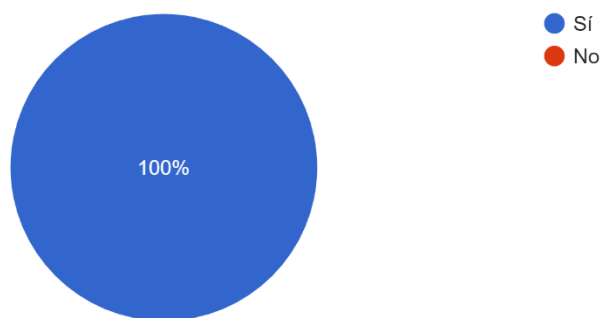


Figura 30. Pregunta tres del formulario de pruebas. Elaboración propia.

¿Los nombres de los botones tiene relación con la tarea a realizar?

3 respuestas

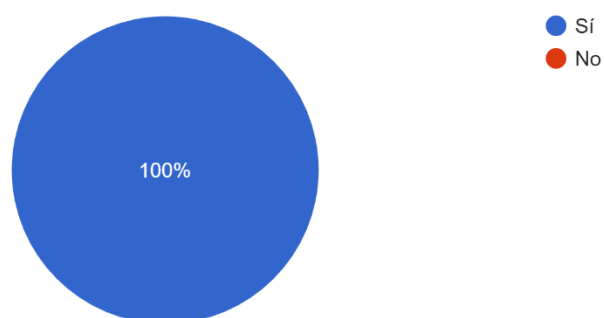


Figura 31. Pregunta cuatro del formulario de pruebas. Elaboración propia.

¿Existe una secuencia para la selección de elementos dentro de las paginas?

3 respuestas

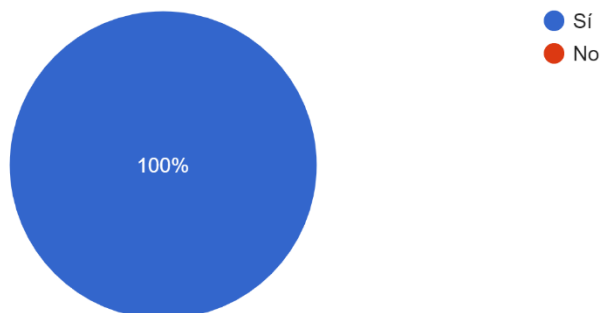


Figura 32. Pregunta cinco del formulario de pruebas. Elaboración propia.

Pruebas funcionales:

¿Al ingresar a la aplicación arrojó algún error?

3 respuestas

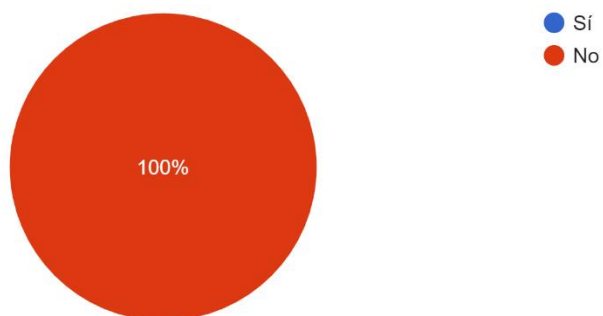


Figura 33. Pregunta uno del formulario de pruebas funcionales. Elaboración propia.

¿Al guardar la información le arrojó algún error?

3 respuestas

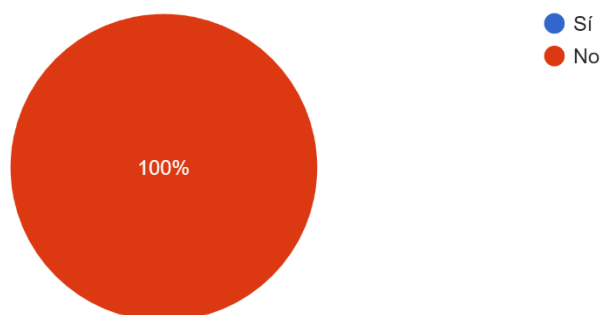


Figura 34. Pregunta dos del formulario de pruebas funcionales. Elaboración propia.

¿Al dar seguimiento al syllabus pudo enviar si el tema fue visto o no?

3 respuestas

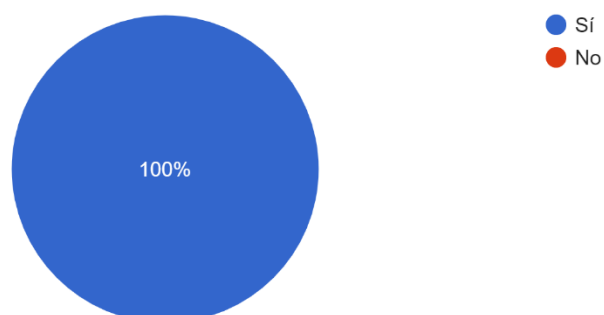


Figura 35. Pregunta tres del formulario de pruebas funcionales. Elaboración propia.

¿La aplicación le permitió diligenciar la encuesta de desempeño del syllabus?

3 respuestas

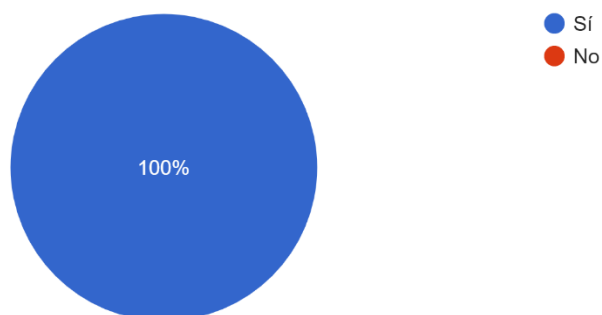


Figura 36 . Pregunta cuatro del formulario de pruebas funcionales. Elaboración propia.

En el perfil administrativo ¿pudo visualizar los informes y descargarlos?

3 respuestas

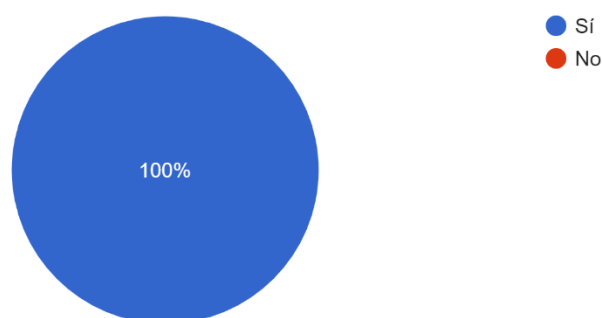


Figura 37. Pregunta cinco del formulario de pruebas funcionales. Elaboración propia.

7.2. Pruebas de usabilidad

Se realizan pruebas de usabilidad por parte de los estudiantes para testear la aplicación, donde se envía un formulario donde realizaran preguntas cerradas sobre el diseño y funcionamiento de la aplicación. Se selecciona un grupo de estudiantes del programa tecnología en desarrollo de software de la jornada nocturna, de dichas pruebas se obtienen los siguientes resultados:

Pruebas de diseño

¿Puede ver en la portada y las demás páginas, la forma en que se navega por el sitio? ¿Se distingue fácilmente?

3 respuestas

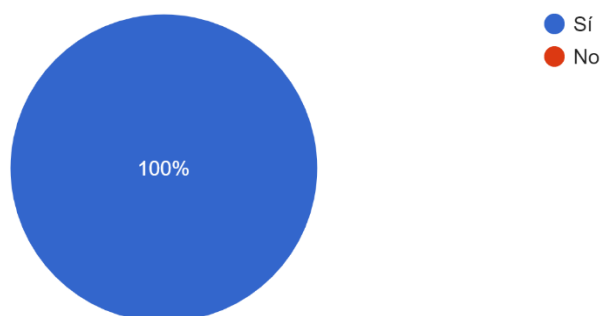


Figura 38. Pregunta uno del formulario de pruebas de diseño. Elaboración propia.

¿A primera vista, le queda claro como es el funcionamiento de la aplicación?

3 respuestas

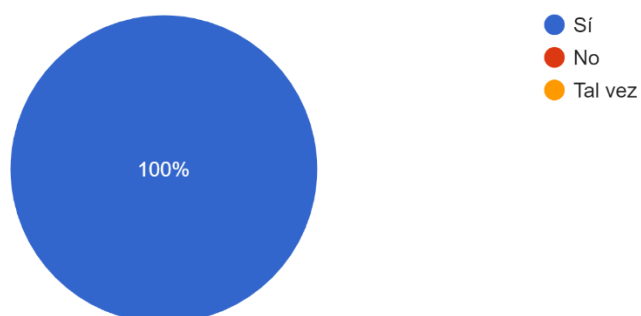


Figura 39. Pregunta dos del formulario de pruebas de diseño. Elaboración propia.

¿Le pareció adecuada la forma en que se muestran los colores, información y botones en la aplicación?

3 respuestas

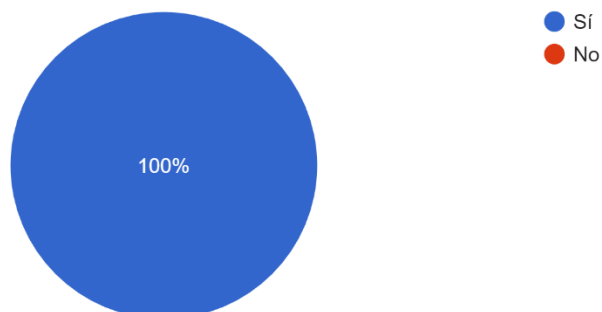


Figura 40. Pregunta tres del formulario de pruebas de diseño. Elaboración propia.

¿El esquema, diseño y estética de la aplicación le parece adecuado?

3 respuestas

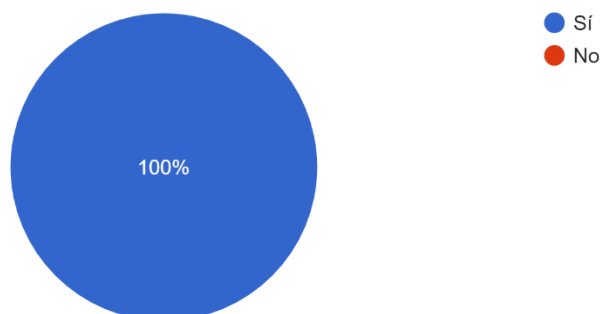


Figura 41. Pregunta cuatro del formulario de pruebas de diseño. Elaboración propia.

Pruebas de funcionamiento:

¿La aplicación presento alguna falla al instalarla o al iniciar sesión?

3 respuestas

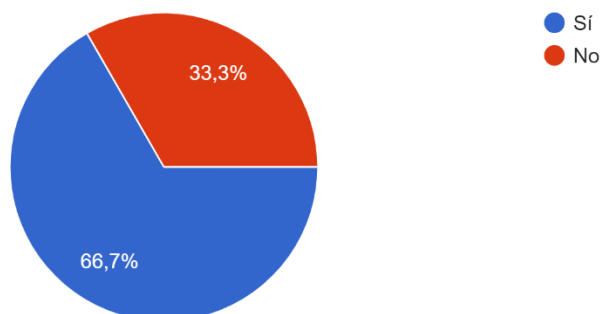


Figura 42. Pregunta uno del formulario de pruebas de funcionalidad. Elaboración propia.

¿Los tiempos de respuesta de cargue de las paginas y envió de la información son adecuados?

3 respuestas

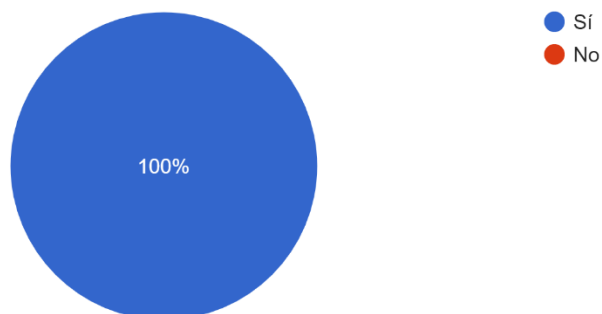


Figura 43. Pregunta dos del formulario de pruebas de diseño. Elaboración propia.

¿Al hacer clic para dar seguimiento al syllabus, pudo enviar la información?

3 respuestas

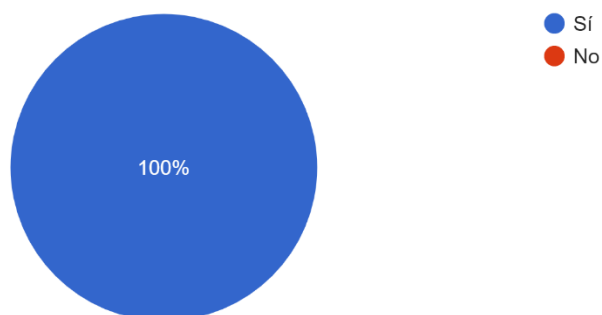


Figura 44. Pregunta tres del formulario de pruebas de diseño. Elaboración propia.

¿Pudo visualizar, responder y enviar la encuesta de desempeño del syllabus?

3 respuestas

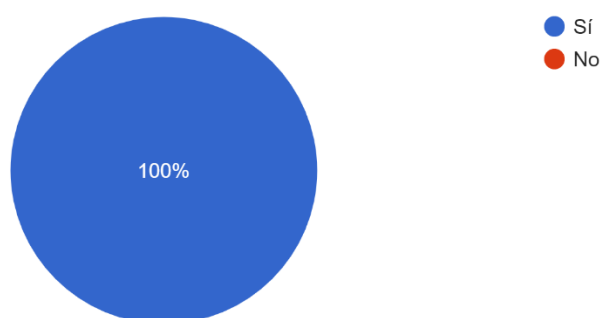


Figura 45. Pregunta cuatro del formulario de pruebas de diseño. Elaboración propia.

Conclusiones

El desarrollo de la aplicación móvil realizada contribuye a realizar el seguimiento del syllabus de las diferentes asignaturas del área disciplinar de programa tecnología en desarrollo de software. Al iniciar la idea de la aplicación se crearon unos requerimientos, los cuales a iniciar el diseño y el análisis de los datos se encontró que sobre pasaban el cronograma que contemplamos para el desarrollo de la aplicación, por cual, se redujeron para se cumplieran los ítems vitales pensados desde un inicio, los cuales son: el seguimiento del syllabus, este fue el primer requerimiento funcional más importante para nosotros y que debíamos cumplir; el segundo, era plasmar la información obtenida por parte del estudiante en un informe para mostrar esta información al personal administrativo del área disciplinar y así dar a conocer las observaciones de los estudiantes.

En la etapa de desarrollo se inició con la creación de la base de datos, con un modelo entidad – relación, posterior a este proceso, se continuo con el desarrollo de la aplicación móvil, donde se utilizaron protocolos HTTP como lo son el get, post y put, para su creación.

De manera local se realizaron las pruebas con resultados óptimos, sin embargo, al realizar las pruebas de despliegue en el servidor, se generaron varios errores, los cuales fueron revisados y corregidos. Después de esto la aplicación quedo con los resultados esperados y cumpliendo con nuestros objetivos iniciales dando a conocer, si el syllabus es cumplido o no y poder enviar estos datos al área correspondiente de una manera donde puede ser analizada, y que si la contemplan al realizar mejoras en los planes de estudio y de igual manera si se encuentra alguna falencia puedan realizar algún correctivo.

Recomendaciones

Los estudiantes del programa tecnólogo en desarrollo de software pueden utilizar el proyecto realizado y ampliarlo a los demás programas de la Uniagustiniana.

Agregar a los informes ya existentes, una presentación de los resultados de los datos obtenidos en gráficas estadísticas según el tipo de resultado.

Que la Uniagustiniana enlace con la base de datos de las estudiantes de los diferentes programas la aplicación móvil creada.

Crear ventana de espera mientras el sistema realiza la petición a la base de datos desde la aplicación móvil.

Crear ventanas emergentes con mensajes instructivos e interactivos de uso en la aplicación móvil.

Referencias

- Android. (s.f.). Introducción a Android Studio. Android. Recuperado de:
<https://developer.android.com/studio/intro>
- Caballero, N. M.(2013). El Syllabus. Utp.ac. Recuperado de:
<http://www.utp.ac.pa/documentos/2016/pdf/el-tecnologico-23-el-syllabus.pdf>
- Cibercolegios. (s.f.). Sobre Cibercolegios. Cibercolegios. Recuperado de:
<https://webpage.cibercolegios.com/pages/cibercolegios.php>
- Classroomliveweb. (s.f.). Classroom live web. classroomliveweb. Recuperado de:
<https://classroomliveweb.com/home/>
- Ecured. (s.f.). Recuperado de: <https://www.ecured.cu/App>
- Eguiluz, J. (s.f.). Introducción a JavaScript. Uniwebsidad. Recuperado de:
<https://uniwebsidad.com/libros/javascript/capitulo-1#:~:text=JavaScript%20es%20un%20lenguaje%20de,mensajes%20de%20aviso%20al%20usuario.>
- Flor Gavilanes, L. C. (2014). Sistema de Seguimiento de Syllabus (Trabajo de titulación, Universidad Tecnológica de Israel). Recuperado de
<http://repositorio.uisrael.edu.ec/bitstream/47000/938/1/UISRAEL%20-%20EC%20-%20SIS%20-%20378.242%20-%20263.pdf>
- Ionicframework (s.f.). Ionic Framework. Ionicframework. Recuperado de:
<https://ionicframework.com/>
- Letelier, P., Penadés, M. (2012). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). (Trabajo de grado). Universidad Politécnica de Valencia. Recuperado de <http://roa.ult.edu.cu/bitstream/123456789/477/1/masyxp.pdf>
- Ministerio de Educación Nacional (20 de abril de 2010). Por el cual se reglamenta el registro calificado de que trata la Ley 1188 de 2008 y la oferta y desarrollo de programas académicos de educación superior [Decreto No. 1295]. Recuperado de
https://www.mineducacion.gov.co/1621/articles-229430_archivo_pdf_decreto1295.pdf
- Muisca. (s.f.). Temas de Estudio. Muisca. Recuperado de:
<https://muisca.co/educolombia/blog/post/1170/temas-de-estudio.html>
- Mysql. (s.f.). Capítulo 1 Información general (s.f.). mysql. Recuperado de:
<https://dev.mysql.com/doc/workbench/en/wb-intro.html>

Oracle. (s.f.). Base de datos. Oracle . Recuperado de: <https://www.oracle.com/co/database/what-is-database.html>

Pacheco, A., Rendón, K. (2018). Herramienta web para instrumento de seguimiento de syllabus (anexo b) para la universidad de guayaquil. caso de estudio FF.CC.MM. FF. (Trabajo de titulación, Universidad de Guayaquil). Recuperado de <http://repositorio.ug.edu.ec/bitstream/redug/32496/1/B-CISC-PTG-1547%20Pacheco%20Romero%20Alexander%20Guillermo%20.%20Rend%c3%b3n%20Peralta%20Kleber%20Danilo.pdf>

Pes, C.(s.f.). Diseño. carlospes. Recuperado de: http://www.carlospes.com/curso_de_ingenieria_del_software/03_01_disenio.ph

PHP. (s.f.). ¿Qué es PHP?. PHP. Recuperado de: <https://www.php.net/manual/es/intro-what-is.php>

Visualstudio. (s.f.). Preguntas frecuentes sobre Visual Studio Code. Visualstudio. Recuperado de: <https://code.visualstudio.com/docs/supporting/faq>

Zabalza, M. (2004). Guía para la planificación didáctica de la docencia universitaria. Trabajo presentado en nomnre de la Universidad de Santiago de Compostela. Recuperado de: <http://www.udc.es/grupos/apumefyr/docs/guiadeguias.pdf>