

Universidad de las Ciencias Informáticas

Facultad 4



**Aplicación móvil para la gestión del control a
clases de los profesores de la Facultad 4**

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autora:

Marlin Remigio Díaz

Tutor:

Ms.C: Yadira Ramírez Rodríguez

La Habana, Noviembre del 2022

“Año 64 del Triunfo de la Revolución”



“La única lucha que se pierde es la que se abandona”

Ernesto Guevara de la Serna

Declaración de Autoría

Declaro ser la autora del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste firmo la presente a los 1 días del mes de Diciembre del año 2022.



Firma de la Autora

Marlin Remigio Díaz



Firma de la Tutora

Ms.C Yadira Ramírez Rodríguez

Dedicatoria

Por difícil que sea el camino a recorrer la meta principal para lograr el éxito siempre será perseverar. Este logro se dedico a mis padres y mi hermana, unos presentes y otros no físicamente hoy aquí, pero sé que todos orgullosos de verme cumplir mi sueño.

Agradecimientos

No me alcanzarían las palabras ni las páginas para agradecer a quienes de una forma u otra han contribuido para que este día se hiciera realidad.

Estando hoy en el final es imposible no mirar atrás y primeramente sentir orgullo yo de estar aquí y lo que eso significa y luego estar consciente y reconocer que sin ustedes no habría sido posible y con ustedes me refiero a mi mamá y mi papá que me han guiado en la vida y me han acompañado en el largo camino de mi educación, por sus deseos de verme como una profesional y contar con la seguridad, apoyo y confianza de ambos. A ustedes que la vida sería demasiado corta para agradecerles y pagarles todo lo que han hecho por mí. Tú Mami que gracias a dios hoy puedo contar con tu presencia, eres sin dudas y sobre todo mi premio mayor,

contigo Hermana que aunque no estas hoy aqui debes saber que fuiste y siempre serás mi mayor ejemplo a seguir.

Papi sabes que más que mío este título tuyo, yo sé que querías estar aquí viéndome hacerlo y no por una pantalla, yo también daría todo porque estuvieras aquí, pero sabes qué? No importa, eso no me limita a decirte hoy gracias por cada regaño, por cada día en el que me decías tienes que seguir aunque el camino sea difícil porque tienes que ser alguien la vida, gracias por ser el gran padre que eres y siempre estar para nosotras. Quiero que sepan los tres que me siento más que afortunada de la familia que me ha tocado.

Además quiero agradecer a mi madrastra Marilen y a mi tía Naty por ser como unas segundas madres, gracias por estar para mí.

A mis tutores Yadira e Ivanié por toda la ayuda que me han brindado en este trayecto tan difícil.

A mi grupo que compartimos juntos estos 5 años en especial a mis mejores amigos Didi y Adrián que siempre han estado para mí en todo momento y que sabemos los tres que este ciclo empieza hoy pero termina el día que se gradúe el último.

A todas mis amistades más cercana por los momentos felices, bonitos y llenos de emociones encontradas que hemos compartido todo este tiempo Arlin, Amelia, Felix, Claudia Mercadet, Dael, Frank, El nene, Daniela, Abe, Ortelio, Claudia.

Y por último a ustedes que les ha tocado soportarme este último año y la verdad me siento más que agradecida con los tres Yosel, Zuleimys y en

especial a ti Aimet que por más difícil que te lo ponga siempre estás ahí para mí y no sé cómo me soportas la verdad, sabes que una hoja no alcanza para agradecerte, de verdad gracias por tanto.

En fin todo aquel que de una forma u otra me han brindado su apoyo para ser hoy una profesional.

Gracias

RESUMEN

El Vicedecanato de Formación es el encargado de realizar la gestión y el control de los procesos docentes-metodológicos en la Facultad 4, entre los que se incluyen los controles a clase (CC). En la Universidad se emplea un sistema de gestión académica denominado Sistema de Gestión Universitaria. Pero este aún no es lo suficientemente eficiente en cuanto a agilidad laboral se trata en el momento en el que se desea informatizar estos resultados de la gestión de los controles a clases (CC), información que es de vital importancia para el trabajo de los vicedecanos docentes y jefes de departamentos. Para realizar este proceso actualmente en la universidad se emplean dos opciones la primera es de forma manual utilizando un modelo físico y la segunda mediante el uso de la sección de control que se encuentra en la plataforma web Akademos, lo cual dificulta notablemente la agilidad para ser llevado a cabo pues el control no se ejecuta en tiempo real ya que para poder hacer uso de este sistema se necesita estar conectado a la red de la universidad y esto trae consigo que en muchas ocasiones haya demora en la entrega de la información al Jefe de Departamento.

Debido a este motivo surge la necesidad de desarrollar una aplicación móvil para la gestión de controles a clases, brindando una propuesta que permita succionar las deficiencias detectadas y mejorar la calidad y agilidad del proceso. Una aplicación que además genere reportes y facilite el trabajo al vicedecano docente.

En el presente trabajo de diploma se explica todo el proceso de selección de las tecnologías y herramientas a utilizar que más se adecuaron a las características que requiere la aplicación. Además se documenta detalladamente el desarrollo de la aplicación móvil propuesta.

Palabras clave: aplicación móvil, control a clase.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación teórica sobre el proceso de gestión de información del control a clases.....	5
1.1 Marco Conceptual	5
1.2 Caracterización del proceso de gestión de información del control a clases	6
1.3 Estudio de aplicaciones informáticas para la gestión de información sobre el control a clases.....	7
1.4 Metodología de desarrollo de software	9
1.5 Lenguajes y herramientas para el modelado de la solución.....	10
1.6 Tecnologías de implementación	11
1.7 Conclusiones del capítulo.....	13
CAPÍTULO 2: Diseño de una aplicación para dispositivos móviles que permita el proceso de gestión de información sobre el control a clases en la Facultad 4	14
2.1 Descripción del contexto del negocio de la propuesta de solución	14
2.2 Requisitos	15
2.3 Análisis y Diseño	21
2.4 Conclusiones del capítulo.....	28
CAPÍTULO 3: Implementación y evaluación de la aplicación para dispositivos móviles que permite mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4	29
3.1 Implementación	29
3.2 Evaluación de la aplicación desarrollada	31
3.3 Pruebas de software.....	31
3.4 Conclusiones del capítulo.....	36
CONCLUSIONES GENERALES	37
RECOMENDACIONES.....	38

REFERENCIAS BIBLIOGRÁFICAS	39
ANEXOS.....	44
Anexo 1: Entrevista realizada a la Jefa del Departamento de Informática de la Facultad 4	44
Anexo 2: Guía de observación para analizar aplicaciones informáticas que gestionan información sobre el control a clases.....	45
Anexo 3: Encuesta aplicada a la jefa del Departamento de Informática de la facultad 4.....	46
Anexo 4: Planilla del Informe de control a clases.....	47
Anexo 5: Historias de Usuarios	49

ÍNDICE DE FIGURAS

Figura 1. Modelo Conceptual del contexto de negocio de la propuesta de solución.....	15
Figura 2. Diseño arquitectónico: Patrón Modelo Vista - Vista Modelo	23
Figura 3. Diseño arquitectónico de la propuesta de solución	24
Figura 4. Diagrama de clases de la propuesta de solución Diagrama de clases	27
Figura 5. Diagrama de Entidad Relación de Base de Datos.....	28
Figura 6. Ejemplo de empleo de estándar de codificación: tamaño máximo de las líneas de código	29
Figura 7. Ejemplo de empleo de estándar de codificación: nombres de las clases y las funciones	30
Figura 8. Ejemplo de empleo de estándar de codificación: declaración de variables	30
Figura 9. Ejemplo de empleo de estándar de codificación: niveles de codificación	30
Figura 10. Comportamiento de las no conformidades en las pruebas unitarias.....	32
Figura 11. Comportamiento de las no conformidades en las pruebas de aceptación	36
Figura 12. Informe de Control a Clases.....	48

Índice de Tabla

Tabla 1. Listado de Requisitos Funcionales	18
Tabla 2. Listado de Requisitos no Funcionales	18
Tabla 3. HU_1: Insertar Control a Clases.....	20
Tabla 4. HU_2: Eliminar Control a Clases	21
Tabla 5. Cantidad de no conformidad por cada iteración	32
Tabla 6. Caso de prueba insertar control a clases	34
<i>Tabla 7. Caso de prueba eliminar control a clases.....</i>	<i>35</i>
Tabla 8. Cantidad de no conformidades por cada iteración.....	35
Tabla 9. HU_3: Modificar control a clases	49
Tabla 10. HU_4: Consultar control a clases	50
Tabla 11. HU_5: Buscar control a clases	51
Tabla 12. HU_6: Listar controles a clases	52
Tabla 13. HU_7: Realizar reportes de controles a clases.....	53

INTRODUCCIÓN

La universidad, en la actualidad, tiene como una de sus más sustanciales metas la de responder a los retos que le sitúa el complejo contexto contemporáneo. Su propósito esencial es lograr la formación integral de los futuros profesionales, los que aportarán con su labor al desarrollo de la sociedad (Tünnermann y Tobón, 2010). En el ámbito universitario se desarrollan diferentes procesos que contribuyen a ese propósito (Mes, 2006): docencia, investigación y extensión universitaria (Guzmán, 2014) o vinculación con la sociedad (Araque, 2012), como también se le denomina, estrechamente relacionados. El proceso docente, dada su naturaleza, tiene la particularidad de integrarlos a todos; por ello existe una preocupación permanente en la comunidad educativa en cuanto a su necesario perfeccionamiento.

Las formas organizativas fundamentales del proceso docente - educativo en la Educación Superior son: la clase, la práctica de estudio, la práctica laboral, el trabajo investigativo de los estudiantes, la autopreparación de los estudiantes, la consulta, la tutoría. Se considera la clase como la forma fundamental para su organización (Bermúdez y Rodríguez, 1996). Según la resolución No. 47/2022 (GOC-2018-O25) sobre el Trabajo Docente y Metodológico para la Educación Superior en la República de Cuba, estableciendo en su artículo 128 dice: “La clase es una de las formas organizativas del proceso docente educativo, que tiene como objetivos la adquisición de conocimientos, el desarrollo de habilidades y la formación de valores e intereses cognoscitivos y profesionales en los estudiantes, mediante la realización de actividades de carácter esencialmente académico”.

En el nivel universitario se requiere que la clase evidencie la creatividad del docente para motivar a los estudiantes por el estudio, la ampliación de sus conocimientos, la participación activa en su adquisición y el establecimiento de sus nexos con la futura profesión (Hernández, 2017). El control se ha convertido en un proceso cotidiano de las instituciones de Educación Superior. El control a los procesos universitarios está dirigido a los resultados a alcanzar, a partir de lo planificado en los objetivos de trabajo a nivel institucional, que comprende procesos docentes y no docentes. Según se establece en la Resolución No. 47/2022, por la que se aprobó el “Reglamento para el Trabajo Docente y Metodológico en la educación superior”, el control al proceso docente educativo es el medio fundamental para conocer la calidad de dicho proceso, evaluar sus resultados y dirigirlo hacia el cumplimiento de sus objetivos.

El control docente presupone dirección, supervisión y control de varios procesos y actividades que forman parte del trabajo docente metodológico, de normativas, resultados docentes, cumplimiento del plan de estudio, entre otras actividades. Para algunos autores el control docente está dirigido a la organización de esfuerzos, a la determinación de objetivos académicos, orientación de las líneas de trabajo docente - metodológico; a la adecuada normativa para alumnos, personal docente, administrativo con la finalidad de establecer objetivos y metas para un trabajo eficiente y exitoso.

Formando parte de las exigencias del trabajo metodológico. Se parte de la idea de que la manera en que se está realizando la observación de las clases es insuficiente pues luego de realizada una entrevista a la jefa del departamento de informática de la Facultad 4 con el objetivo de detectar las deficiencias de este proceso de gestión de información del control a clases en la facultad y universidad, en la misma se detectó como deficiencia principal y que más afecta el proceso la pérdida de tiempo que conlleva hoy en día realizar la acción pues se realiza de forma manual, además de la pérdida accidental o deterioro de estos documentos.

Una alternativa para corregir estas deficiencias y que el proceso se realice de una forma ágil es la utilización de los dispositivos móviles orientada a tareas con este objetivo, facilitando el trabajo de las personas que se dedican al control docente. Aunque el empleo de los dispositivos móviles va en aumento, no se dispone de un mecanismo que permita crear y gestionar contenidos mediante una interfaz amigable que posea los campos de control a clases.

Sobre la base de los elementos expuestos anteriormente se formula el siguiente **problema de investigación**: ¿Cómo mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4?

Para la realización de la investigación se define como **objeto de estudio**: el proceso de gestión de información sobre el control a clases y el **campo de acción** se encuentra enmarcado aplicaciones informáticas para la gestión de información sobre el control a clases.

Para dar solución al problema planteado, se define como **objetivo general**: Desarrollar una aplicación para dispositivos móviles que permita agilizar el proceso de gestión de información sobre el control a clases en la Facultad 4.

Para dar cumplimiento al objetivo general antes mencionado, se dará cumplimiento a los siguientes **objetivos específicos**:

1. Elaborar la fundamentación teórica sobre el proceso de gestión de información del control a clases.

2. Diseñar una aplicación para dispositivos móviles que permita mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4.
3. Implementar una aplicación para dispositivos móviles que permita mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4.
4. Evaluar la aplicación para dispositivos móviles que permita mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4.

Para el desarrollo de las tareas científicas se han combinado diferentes métodos teóricos y empíricos de la investigación en la búsqueda y procesamiento de la información. Estos son:

Métodos teóricos:

- **Analítico-Sintético:** se empleó en la búsqueda de los elementos esenciales para el modelo de control docente a través de dispositivos móviles. También en el análisis de la estructura de sistema operativo Android para valorar, conocer sus particularidades y lograr comprender su funcionamiento.
- **Inductivo-Deductivo:** se aplicó para la determinación de las generalidades y se parte del análisis de casos particulares, para arribar a razonamientos que permitan la fundamentación teórica y elaboración del sistema que se desea.

Métodos empíricos:

- **Entrevistas:** (Anexo 1) se evidenció en conversaciones planificadas para obtener información con la Jefa del Departamento de Informática de la Facultad 4 sobre cómo se realiza el proceso de control a clases en la facultad, qué información se procesa sobre estos controles y qué deficiencias persisten en la actualidad en la ejecución de dicho proceso. Además, permitió realizar el levantamiento de requisitos, así como un mejor entendimiento acerca de la tecnología a utilizar para el desarrollo de la propuesta de solución.
- **Observación:** (Anexo 2) fue empleada en los distintos momentos de la investigación para la recogida de la información precisa, real y confiable que ayuden a comprender lo principal de la problemática. Al mismo tiempo esto permitió el planteamiento del problema, dando paso a enmarcar el objeto de estudio y el campo de acción, propiciando enfocar la investigación hacia lo que se necesita alcanzar y cómo alcanzarlo.
- **Encuesta:** (Anexo 3) identificar funcionalidades y características que debe poseer la propuesta de solución.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica sobre el proceso de gestión de información del control a clases

En este capítulo se definen conceptos importantes para la comprensión de la investigación, se describen las principales herramientas informáticas empleadas. Se realiza un estudio de las metodologías de desarrollo de software y los sistemas homólogos en cuanto a su objetivo.

Capítulo 2: Diseño de una aplicación para dispositivos móviles que permita mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4

A lo largo de este capítulo se especifican los requisitos funcionales y no funcionales necesarios para lograr que el módulo de procesamiento estadístico para el apoyo a la toma de decisiones se integre de forma satisfactoria al motor de búsqueda Orión, los elementos fundamentales del diseño y de la arquitectura a tener en cuenta para la implementación del sistema, auxiliado por el modelado de diagramas.

Capítulo 3: Implementación y evaluación de la aplicación para dispositivos móviles que permite mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4

En este capítulo se muestra el modelo de implementación como resultado del diseño anteriormente realizado. Se describen las pruebas a utilizar y los resultados obtenidos de estas, con el objetivo de comprobar el correcto funcionamiento de los sistemas de recuperación de información y el procesamiento estadístico para la toma de decisiones.

CAPÍTULO 1: Fundamentación teórica sobre el proceso de gestión de información del control a clases

Introducción

En el presente capítulo se muestran elementos sobre la fundamentación teórica de la investigación. Se realiza una caracterización del proceso de gestión de información del control a clases. Contiene un estudio sobre aplicaciones informáticas para el control a clases. Se define la metodología de desarrollo de software utilizada en la elaboración de la propuesta de solución. Se evidencia los lenguajes y herramientas empleados en el modelado y desarrollo de una aplicación móvil para la gestión del control a clases de los profesores de la Facultad 4.

1.1 Marco Conceptual

Para lograr una mejor comprensión de la investigación, se abordan un conjunto de conceptos necesarios que están estrechamente relacionados con el dominio del problema.

Aplicación móvil

Las aplicaciones móviles son programas diseñados para ser ejecutados en teléfonos, tabletas y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades (Bernal,2021).

Telefonía móvil

El teléfono móvil es un dispositivo inalámbrico electrónico basado en la tecnología de ondas de radio, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Su principal característica es su portabilidad, ya que la realización de llamadas no es dependiente de ningún terminal fijo y no requiere ningún tipo de cableado para llevar a cabo la conexión a la red telefónica. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea, agenda, juegos, cámara fotográfica, acceso a Internet, reproducción de video y reproductor mp3 (IFT, 2015).

1.2 Caracterización del proceso de gestión de información del control a clases

1. Según el artículo 186 de la Resolución 47/2022. Los tipos de actividades metodológicas que se planifican en los diferentes niveles organizativos del proceso docente educativo pueden ser objeto de control por parte de los directivos académicos acompañados por profesores responsables de colectivos metodológicos, con el propósito de verificar la calidad con que se desarrollan. A tales fines se deben considerar entre otros aspectos, los siguientes:

- La preparación de la actividad.
- El cumplimiento de los objetivos propuestos y su pertinencia.
- El dominio del tema.
- La calidad en el tratamiento del contenido.
- La preparación y participación de los participantes.
- La profundidad de los análisis.
- Las valoraciones y debates realizados.
- La propuesta de acuerdos, en caso necesario.

Los controles a las actividades metodológicas se incluyen en la planificación del trabajo metodológico y sus resultados son recogidos en un acta por el responsable del control.

2. En una entrevista realizada a la Jefa del Departamento de Informática de la Facultad 4 se pudo conocer las características que tiene el proceso de gestión de información del control a clases en la facultad y universidad, relacionadas a continuación.

Un profesor en correspondencia con su categoría docente puede ser controlado durante el curso con la frecuencia siguiente:

- Recién Graduado en Adiestramiento (RGA) que dispone de una licencia para impartir clases se controla tantas veces como sea necesario.
- Instructor 2 veces.
- Asistente 2 veces.
- Auxiliar 1 veces.
- Titular 1 veces.

El controlador por cada clase que controla debe redactar un informe (Anexo 4) en el que se detalla la siguiente información:

- Nombre y Apellidos del docente controlado y su categoría docente.

- Nombre y Apellidos del controlador y su categoría docente.
- Asignatura, Tipo de clase y Título de la clase
- Grupo que recibe la clase
- Los principales logros de la clase
- Las deficiencias detectadas
- Las recomendaciones realizadas
- Le evaluación que recibe el profesor
- La fecha del control realizado
- La firma del controlador y del profesor controlado

La entrevista permitió detectar un conjunto de deficiencias que son necesarias corregir para el correcto funcionamiento de este proceso, destacándose entre ellos como el hallazgo principal la pérdida de tiempo que conlleva la realización manual del Informe de control a clases y su posterior informatización en el Sistema de Gestión Académica, así como la pérdida o extravío del informe.

Comprendido el proceso de gestión de información del control a clases en la Facultad 4, la autora de la investigación realizó un estudio de aplicaciones informáticas existentes para verificar si alguna solucionaba los problemas que dan origen al trabajo de diploma. En el siguiente epígrafe se describen las aplicaciones analizadas.

1.3 Estudio de aplicaciones informáticas para la gestión de información sobre el control a clases

En el siguiente epígrafe se efectúa un análisis de las aplicaciones móviles existentes para la gestión de control a clases, luego de un estudio fueron analizados Control L, Tabalive gestión de aula, Additio App ya que fueron los más relevantes que se encontraron, en busca de elementos y funcionalidades que pudieran ayudar a resolver el problema planteado. A continuación, se detallan algunas de las características de estos sistemas.

Aplicación Control-L

Es una plataforma que permite una gestión más ágil ya que el profesor desde el móvil, es el que realiza en tiempo real la mayor parte del trabajo de gestión: registro y evaluación de clases y exámenes, presentar a examen. La información se comparte instantáneamente con la administración de la autoescuela y con los alumnos (Control-L, 2016).

Aplicación Additio App

Additio App funciona como un cuaderno de notas para que los docentes planifiquen el curso, controlen asistencias y lleven un seguimiento de sus notas. Sus características incluyen: creación de tutorías de aviso, personalización de horarios y visualizar de forma semanal las tareas correspondientes. Tiene dos versiones: *Additio plus*, para docentes, y *Additio for Schools*, para centros escolares. Y la opción de *Edvoice*, una aplicación que permite crear *stories* y chats con los que comunicarse con la comunidad educativa (Educación 3.0, 2022).

Akademos

El Sistema de Gestión Universitaria fue creado con el objetivo de mantener la mayor cantidad de información legalmente manejada por la universidad, centralizada para el uso de los encargados de manejarla constantemente. Muestra una interfaz común para todos los usuarios muy intuitiva para el intercambio con la misma, brinda información necesaria para todas las facultades de la Universidad. Está compuesto por diferentes módulos entre los que se encuentran: Postgrado, Cooperación, Residencia, Ingreso, Investigación, Producción, Laboratorios, Biblioteca, Desarrollo, Tecnología, Extensión, Teleformación, Egreso y Pregrado siendo en este último donde se realizan los controles a clases. Para realizar este proceso actualmente en la universidad se emplean dos opciones la primera es de forma manual utilizando un modelo físico y la segunda mediante el uso de la plataforma web, lo cual dificulta notablemente la agilidad para ser llevado a cabo pues el control no se ejecuta en tiempo real ya que para poder hacer uso de este sistema se necesita estar conectado a la red de la universidad y esto trae consigo que en muchas ocasiones haya demora en la entrega de la información al Jefe de Departamento, deterioro del documento o incluso pérdida de la información.

Desde el punto de vista de obtención de información centralizada para los vicedecanatos de formación de las facultades es muy necesario el uso de esta herramienta, pero cabe señalar que es necesario mejorar, expandir nuevas alternativas con el fin de hacer más ágil la gestión de los controles a clases, los cuáles son tan necesarios para el trabajo metodológico y a la hora de generar informes completos para la máxima dirección de la Universidad. Es por ello que se hace imprescindible implementar la propuesta.

1.3.1 Análisis comparativo de las aplicaciones informáticas

Luego de realizar un análisis en los sistemas seleccionados, observando sus características y contribuciones, fue posible resumir que no cumplían del todo con las necesidades de la situación

problemática; aun así se utilizaran algunas funcionalidades de estos. De los sistemas *Contro_L* y *Adicto_App* se toma como referencia la proyección y enfoque en cuanto a organización a la hora de realizar sus controles. Del sistema Akademos partirá la idea principal para la implementación de la aplicación móvil para la gestión de control a clases de los profesores de la Facultad 4 ya que su objetivo principal es agilizar la ejecución de este proceso haciendo uso de la telefonía móvil.

1.4 Metodología de desarrollo de software

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (Domain, 2021). Es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tiene como objetivo apoyar a los desarrolladores en la creación de un nuevo software (Zambrano, 2013).

Se puede decir que la metodología de desarrollo es un proceso de construcción del *software* con el objetivo de trazar una línea secuencial de las actividades que se deben realizar, haciendo uso de métodos, modelos, herramientas y gestión de las aptitudes del equipo de desarrollo para que todo esto aporte a la calidad final del producto.

Para el desarrollo de la solución se usará la variación de la metodología de Proceso Unificado Ágil (AUP por sus siglas en inglés), por ser un producto que se tiene para la comunidad universitaria y estar aún expenso en su desarrollo a seguir iterando, se decide seguir el trabajo de esta metodología por ser la utilizada en la universidad en los proyectos.

1.4.1 Metodología de desarrollo de software Variación de AUP para la UCI

La metodología utilizada en la universidad es una variante de AUP (*Agile Unified Process*, Proceso Ágil Unificado). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software, usando técnicas ágiles. Se adapta al ciclo de vida definido para la actividad productiva de la institución, logra estandarizar los productos de trabajo elaborados en los diferentes centros de desarrollo de la universidad. Propone tres fases (Rodríguez, 2015):

- Inicio: en esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.

- Ejecución: en esta fase se recogen las actividades que desarrolla AUP de elaboración, construcción y transición. Se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura.
- Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

En la presente investigación se empleó en la fase de Ejecución, pues la misma propicia que se ejecuten las actividades necesarias para realizar el software; además aborda las disciplinas: Requisitos, Análisis y diseño, Implementación, Pruebas internas y Pruebas de aceptación (Rodríguez, 2015). De los 4 escenarios que propone la metodología para la descripción de requisitos se emplea el cuarto ya que es factible aplicarlo a proyectos bien definidos; también se ha de tener en cuenta que el cliente siempre acompaña al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

1.5 Lenguajes y herramientas para el modelado de la solución

En la presente investigación se emplea como lenguaje de modelado UML y como herramienta para el modelado Visual Paradigm, ambos se describen a continuación:

Lenguaje Unificado de Modelado (UML) v8.0

El Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) es el lenguaje estándar especificado por el *Object Management Group* (OMG) para visualizar, especificar, construir y documentar los artefactos de un sistema informático, incluyendo su estructura y diseño. Utiliza un conjunto de símbolos y notaciones para representar gráficamente los diversos componentes que forman parte de la arquitectura de software. Permite el modelado de procesos de negocio y el modelado de requisitos apoyándose en el análisis orientado a objetos (Object Management Group, 2015).

Herramienta para el modelado de la solución

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) se utilizan para una rápida construcción de aplicaciones. Permite construir diagramas como el diagrama de proceso del negocio y el diagrama de despliegue. Se considera muy completa y fácil de usar, con soporte multiplataforma. Permiten integrar durante todo el proceso de ciclo de vida del proyecto aspectos como: la generación de interfaces entre el análisis y el diseño y la generación del código a partir del diseño. En la elaboración de los diferentes diagramas y modelos empleados en la investigación se utilizó Visual Paradigm (Méndez, 2020).

Visual Paradigm v5.0

Visual Paradigm es una herramienta CASE para el modelado UML que permite analizar, diseñar, codificar, probar y desplegar. Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Visual Paradigm Internationals, 2018).

1.6 Tecnologías de implementación

Para el desarrollo de la propuesta de solución se hace necesario la definición de las herramientas, tecnologías y metodología que se emplearán para la realización de la aplicación móvil para el control docente.

1.6.1 Lenguajes de programación

Android

Entorno de desarrollo basado en *IntelliJ IDEA*. Posee distintos componentes que ayudan en la tarea de la construcción de aplicaciones; sistema de construcción basado en *Gradle*, la construcción de variantes y múltiples archivo APK, como también plantillas de código. Contiene una interfaz de usuario que es construida o diseñada previamente, con variados modelos de pantalla, donde los elementos existentes pueden ser desplazados. Adicionalmente se abarca depuradores para emuladores y la posibilidad de trabajo con *Logcat*. Herramienta que mejora el rendimiento, facilita el uso y no producen problemas de compatibilidad con las diferentes versiones. Facilita la reutilización de código, de recursos, posibilitando además configurar, extender y personalizar el proceso (Nobbot, 2021).

Kotlin

Kotlin es un lenguaje de programación pragmático pensado para funcionar con Máquina Virtual de Java (JVM) y Android. Además, puede ser compilado a código fuente de JavaScript. Se caracteriza por una perfecta combinación de características claramente orientadas a la funcionalidad durante la programación, centrándose en la seguridad, la claridad y la interoperabilidad. *Kotlin* destaca por varias características que no solo simplifican la lectura del código sino el propio desarrollo de este. Es por ello que las ventajas de *Kotlin* respecto a Java a la hora de desarrollar aplicaciones móviles ha hecho que este lenguaje se haga cada vez más popular (Guimerá, 2021).

1.6.2 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C# y otros. En algunos lenguajes, puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Delgado, 2021).

1.6.3 Repositorio de dependencias

Maven

Maven es una herramienta de gestión y comprensión de proyectos. Maven ofrece a los desarrolladores un marco de trabajo para construir todas las etapas del ciclo de vida de una aplicación. El equipo de desarrollo puede automatizar la infraestructura necesaria para la creación de proyectos en poco tiempo. Maven utiliza un esquema de directorio estándar y una construcción de etapas clásicas de ciclo de vida de un proyecto. En el caso en el que el desarrollo de un proyecto, sea realizado por varios equipos de desarrollo, Maven puede configurar la forma de trabajar basándose en estándares en muy poco tiempo. Como la mayoría de las configuraciones de proyecto son simples y reutilizable, Maven facilita la vida a los desarrolladores proporcionando de forma automática: la creación de informes, validaciones, compilación y pruebas de configuraciones de forma automatización (García, 2015).

1.6.4 Herramienta para la gestión de dependencias

Gradle

Gradle es una herramienta de automatización de la construcción de nuestro código que bebe de las aportaciones que han realizado herramientas como *ant* y maven pero intenta llevarlo todo un paso más allá. Para empezar se apoya en *Groovy* y en un DSL (*Domain Specific Language*, Lenguaje de dominio específico) para trabajar con un lenguaje sencillo y claro a la hora de construir el *build* comparado con Maven. Por otro lado dispone de una gran flexibilidad que permite trabajar con ella utilizando otros lenguajes y no solo Java. Dispone por otro lado de un sistema de gestión de dependencias sólido. (Álvarez, 2015).

1.6.5 Herramienta para el control de versiones de código

GitLab

GitLab es un repositorio de gestión de proyectos y, como se puede deducir desde el nombre, está construido sobre *Git*. Es decir, proporciona el código para generar un servidor y gestionar los clientes, sus operaciones y los servicios que ofrece. A través de *GitLab* puedes administrar grupos, personas y los permisos que quieras que tengan los usuarios dentro de los grupos o proyectos a los que pertenezcan. Si estás buscando cómo aumentar la productividad laboral en tu empresa, esta herramienta puede ayudarte. También, te permite llevar a cabo un seguimiento del estado actual y del histórico de los proyectos, logrando ver todos los cambios y modificaciones producidas en el tiempo de desarrollo. Teniendo presente que, *GitLab* es un *software* libre que puedes descargar e instalar en cualquier servidor, permitirá que lo uses en tu equipo personal, de manera profesional y en los ordenadores de tu organización sin ningún costo (Gómez, 2022).

1.7 Conclusiones del capítulo

El análisis del marco teórico sobre el proceso de gestión de información sobre el control a clases y el estudio de los principales conceptos asociados al problema planteado, permitieron establecer las bases para el desarrollo de la investigación y conocer las características del objeto de estudio. A partir de la caracterización y comparación de varias aplicaciones informáticas para la gestión de información sobre el control a clases se demostró la necesidad de desarrollar una aplicación móvil que utilice las mejores funcionalidades que estas aplicaciones ofrecen, pues estas no resuelven la totalidad de las insuficiencias. En el desarrollo de la propuesta de solución se definió el empleo de herramientas informáticas en su mayoría libres para garantizar la soberanía tecnológica, y la utilización de la metodología de desarrollo de software Variación de AUP para la UCI.

CAPÍTULO 2: Diseño de una aplicación para dispositivos móviles que permita el proceso de gestión de información sobre el control a clases en la Facultad 4

Introducción

Este capítulo aborda los aspectos técnicos de la investigación referentes al análisis y diseño de una aplicación para la gestión de información del control a clases. Contiene una descripción del contexto de dominio de la propuesta de solución. Se definen las fuentes empleadas en la obtención de los requisitos y las técnicas aplicadas para su elicitación. Muestra la especificación de requisitos de software, su descripción mediante historias de usuario y la validación de los mismos. Se realiza el diagrama de clases del diseño, en el que se representan las restricciones de las funcionalidades y el empleo de los patrones del diseño. Se presenta el modelo de datos que describe la estructura de persistencia de los datos relacionados con el control a clases y el diseño arquitectónico definido para la propuesta de solución.

2.1 Descripción del contexto del negocio de la propuesta de solución

La descripción del contexto del negocio presentada en el epígrafe se realizó utilizando un modelo conceptual. El mismo le permitió a la autora de la investigación comprender el funcionamiento del dominio del negocio para ser informatizado.

Modelo conceptual

Un modelo conceptual contiene la descripción de cómo se relacionan los conceptos que intervienen en el negocio y sirve para representar el problema de manera gráfica. Los conceptos que en este modelo aparecen, son una representación de elementos del mundo real y no de componentes de software (Dalvik, 2018).

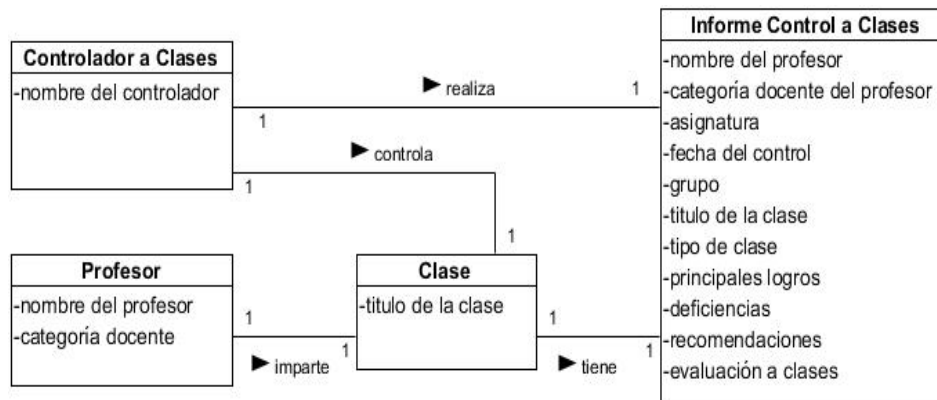


Figura 1. Modelo Conceptual del contexto de negocio de la propuesta de solución

Fuente: Elaboración propia

Las definiciones que se muestran a continuación permiten comprender el significado que tienen los conceptos representados en el modelo conceptual del contexto del negocio informatizado.

Conceptos del contexto del negocio:

- Controlador a Clases: es la persona que realiza el control a una clase.
- Profesor: es la persona que es controlada.
- Clase: es una forma organizativa fundamental del proceso docente - educativo en la Educación Superior para la formación de los estudiantes, es impartida por el Profesor y evaluada por un Controlador a Clases.
- Informe Control a Clases: es el documento que redacta el Controlador a Clases con los aspectos evaluados.

Una vez comprendido el contexto del negocio informatizado, la autora de la investigación procedió a obtener los requisitos de software de la solución, así como su especificación.

2.2 Requisitos

La disciplina de requisitos es una de las más importantes en el proceso de desarrollo de software; consiste en desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez, 2015).

2.2.1 Obtención de requisitos

En la obtención de requisitos de la propuesta de solución se emplearon diferentes fuentes y técnicas, descritas a continuación.

Fuentes para la obtención de requisitos

Uno de los elementos más importantes del proceso de desarrollo del software es la obtención de los requisitos, debido a que ayuda a conciliar conflictos de intereses entre los involucrados, y determinar qué tipo de software se desea desarrollar (Pérez, 2017). En este proceso intervienen diferentes fuentes que permiten identificar los requisitos que forman parte de una aplicación informática.

Durante esta etapa de la investigación se tuvieron en cuenta como fuentes de obtención de requisitos:

- Caracterización del proceso de gestión de información del control a clases (Epígrafe 1.2).
- Aplicaciones informáticas para la gestión de información del control a clases (Epígrafe 1.3).
- Descripción del contexto de dominio de la propuesta de solución (Epígrafe 2.1).
- Jefa de Departamento de Informática Facultad 4.

Técnicas de identificación de requisitos

Las técnicas de identificación de requisitos de software facilitan la identificación de las funcionalidades y características de un software en correspondencia con las necesidades de negocio de los clientes. Se utilizan para recolectar la información necesaria en la obtención de los requisitos de una aplicación, mediante la investigación de aspectos generales para posteriormente ser especificados con un mayor detalle. Estos mecanismos deben ser orientados para cubrir la información que se requiere capturar (Bescos, 2019). Las técnicas empleadas en la investigación se describen a continuación.

Entrevista

Es una técnica ampliamente utilizada para la recolección de opiniones, criterios o descripciones sobre diferentes actividades. Se lleva a cabo mediante conversaciones estructuradas donde es fundamental que la relación con el cliente este basada en la confianza para dar a conocer la información de la manera más detallada (Guerra, 2018). La técnica se aplicó a través de una entrevista realizada a la Jefa del Departamento informática de la Facultad 4 (ver Anexo 3) para conocer cuáles son las tecnologías compatibles para el desarrollo de la aplicación, así como los requerimientos y las características que debe tener la aplicación.

Observación

Por medio de esta técnica se obtiene información sobre la forma en que se efectúan las actividades. Esta permite observar la manera en que se llevan a cabo los procesos y verificar que realmente se sigan todos los pasos especificados. Los observadores experimentados saben qué buscar y cómo evaluar la relevancia de lo que observan (Guerra, 2018). Se aplicó esta técnica a través de la observación de cómo se aplica un control a clases en la facultad y los diferentes aspectos que se miden para luego emitir una calificación (ver anexo 4), puesto que son parte de la fundamentación de la aplicación a desarrollar.

Desarrollo de prototipos

Los prototipos consisten en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Permiten a los usuarios experimentar para ver cómo este ayuda a su trabajo, fomentan el desarrollo de ideas que desembocan en requerimientos, mejoran las especificaciones de requerimientos, identifican discrepancias entre los desarrolladores y los usuarios y permite formalizar la aceptación previa por parte del cliente de los requisitos del proyecto (Guerra, 2018). Se aplicó la técnica de desarrollo de prototipos para conocer cuáles son las características que debían tener las interfaces de la aplicación desarrollada. Los prototipos elaborados fueron revisados, y se concluyó que se debía contar con estos prototipos para la representación visual de la propuesta de solución.

La aplicación de las técnicas descritas permitió comprender con profundidad el problema de la investigación y facilitó la identificación de las funcionalidades y características de la propuesta de solución, definidas en el siguiente epígrafe.

2.2.2 Especificación de requisitos de software

La especificación de requisitos de software es una descripción del sistema de software que se desarrollará. Se basa en la definición de las necesidades de las partes interesadas. Incluye requisitos funcionales y no funcionales, y puede incluir un conjunto de casos de uso que describen las interacciones del usuario que el software debe proporcionar para una perfecta interacción (Arias, 2019).

Requisitos funcionales

Los Requisitos funcionales (RF) expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento) (Martin, 2019).

Las entrevistas realizadas arrojaron un total de 7 requisitos funcionales, que a continuación se presentan.

Tabla 1. Listado de Requisitos Funcionales

Fuente: Elaboración propia

Número	Nombre del requisito	Descripción del requisito
RF1	Insertar control a clases	Permite crear un control a clases
RF2	Modificar control a clases	Permite modificar los datos de un control a clases
RF3	Eliminar control a clases	Permite eliminar un control a clases
RF4	Consultar control a clases	Permite mostrar la información de un control de clases
RF5	Buscar control a clases	Permite buscar un control a clases
RF6	Listar controles a clases	Permite listar los controles a clases realizados
RF7	Realizar reporte de controles a clases	Realiza un reporte de los controles a clases realizados

Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades que hacen al producto atractivo, usable, rápido y confiable. Son los encargados de darle el acabado al producto para que esté listo para usarse (Pmoinformatica, 2015). Se identificaron un total de 7 requisitos no funcionales, agrupados en 5 categorías que a continuación se presentan.

Tabla 2. Listado de Requisitos no Funcionales

Fuente: Elaboración propia

Número	Requisito	Descripción
RnF 1	Requisito de Software	El dispositivo móvil debe tener sistema operativo Android 5.1 o superior para poder utilizar la aplicación móvil.
RnF 2	Requisito de Hardware	El dispositivo móvil debe contar como mínimo con 3 megas de almacenamiento que la aplicación sea instalada y funcione correctamente.
RnF 3	Requisito de diseño e Implementación	Como lenguaje de programación para el desarrollo de la aplicación se deberá utilizar Kotlin en su versión 1.4.

RnF 4	Requisito de Usabilidad	La aplicación móvil debe permitir acceder a sus diferentes interfaces con una profundidad máxima de 2 clics.
RnF 5		La aplicación debe adaptarse a diferentes resoluciones de pantalla.
RnF 6		La aplicación debe presentar una interfaz de usuario simple para facilitar el manejo a los usuarios.
RnF 7	Requisito de Rendimiento	El tiempo de carga de datos no puede superar los 3 milisegundos de respuestas.

Para describir las características de los requisitos funcionales definidos en el epígrafe 2.2.2, así como sus restricciones en el contexto del negocio informatizado, se utilizó el producto de trabajo Historia de Usuario. En el epígrafe 2.2.3 se presenta un ejemplo de este artefacto de la ingeniería de software.

2.2.3 Descripción de requisitos funcionales

La descripción de los requisitos funcionales de la propuesta de solución se realizó mediante Historias de usuario.

La Historia de Usuarios (HU) permite registrar los requerimientos de los clientes según el negocio y son utilizadas para poder realizar la estimación de cada una de las iteraciones durante la fase de planificación. Son escritas por el equipo de trabajo en conjunto con los clientes en base a lo que se estima que es necesario para el sistema (Vergara, 2021).

Tabla 3. HU_1: Insertar Control a Clases

Fuente: Elaboración propia


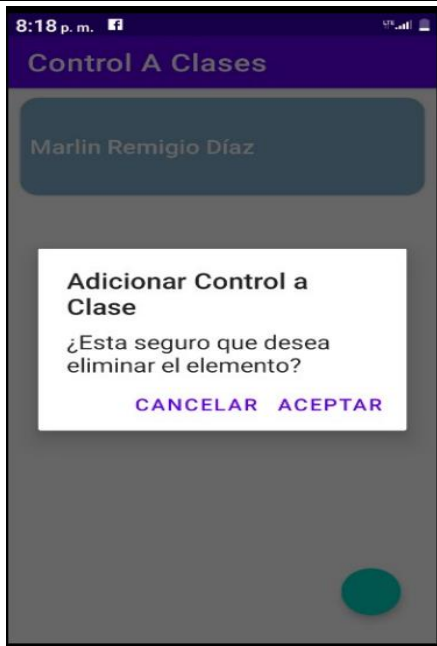
Historia de usuario	
Número: HU_1	Nombre: Insertar control a clases.
Prioridad en negocio: Alta	
<p>Descripción: La aplicación permite al usuario crear un control a clases. El usuario selecciona la opción de “Adicionar control a clases” del menú principal. Luego aparecen los campos para insertar nombre de la persona a la que se le va a realizar el control, grupo, título de la clase, nombre del controlador. Seleccionar la categoría docente del controlado, categoría docente del controlador, asignatura y tipo de clase. En caso de haber dejado alguno de los campos vacíos se enviara un mensaje notificando al usuario que existen campos por completar “Existen campos vacíos”. Una vez llenados los campos se selecciona la opción de “Adicionar” en el menú inferior. En caso de que el usuario se arrepienta en la creación de una planificación se cuenta con la opción “Cancelar” también en el menú inferior.</p>	
Prototipo:	

Tabla 4. HU_2: Eliminar Control a Clases

Fuente: Elaboración propia

Historia de usuario	
Número: HU_2	Nombre: Eliminar control a clases.
Prioridad en negocio: Media	
<p>Descripción: Permite al usuario eliminar un control a clases. El usuario va al listado de controles realizados y selecciona el control a clases que desea eliminar y lo deja presionado y a continuación se muestra un mensaje de confirmación ¿Está seguro que desea eliminar el elemento?, luego en la parte inferior se muestran los botones de cancelar y aceptar q deberá seleccionar para confirmar su petición.</p>	
Prototipo:	

2.3 Análisis y Diseño

En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Se modela el sistema, su forma y su arquitectura para que soporte todos los requisitos, incluyendo los requisitos no funcionales (Rodríguez, 2015). En los epígrafes siguientes se muestran las actividades desarrolladas en esta disciplina, así como los productos de trabajo elaborados en el proceso de desarrollo en la aplicación móvil para el control docente a profesores de la facultad 4.

2.3.1 Diseño arquitectónico

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”. Consiste en la estructura o sistema de estructuras, que comprenden los elementos de software, las propiedades externas visibles de esos elementos y la relación entre ellos. Es una vista del sistema que incluye sus componentes principales, así como la conducta de esos componentes y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema (Lucidchart, 2022). El diseño arquitectónico de la propuesta de solución fue elaborado con el patrón arquitectónico Modelo Vista - Vista Modelo, descrito a continuación.

Patrón arquitectónico Modelo Vista - Vista Modelo

“El patrón MVVM permite diferenciar la lógica de negocio de la interfaz gráfica, facilitando las actividades realizadas en las pruebas, despliegue y mantenimiento del proyecto”.

Existen tres capas:

Modelo: Contiene las entidades que describen el modelo de la aplicación dentro del cual se encuentra:

El modelo de datos y la lógica del negocio. Aquí se definen los atributos que utiliza la app.

Modelo de vista: Contiene toda la lógica de presentación, además de complementar las propiedades y comandos que definen las funcionalidades de la app.

Fuentes: Enlazan con la vista, por lo tanto, se encarga de comunicar el modelo con la vista. En esta capa, se encuentran las interfaces y métodos que agregan la funcionalidad de cada clase implementada en el modelo.

Vista: Más conocida como “*User Interface*” (UI), define la forma en como la información y las funcionalidades se muestran gráficamente. Se tiene la posibilidad de definir la estructura con marcos de trabajo para desarrollar móviles, como *Xiaomi*, *Foms*, *Roact Native*, *Flutter*, entre otros. (Guerra y López, 2020).

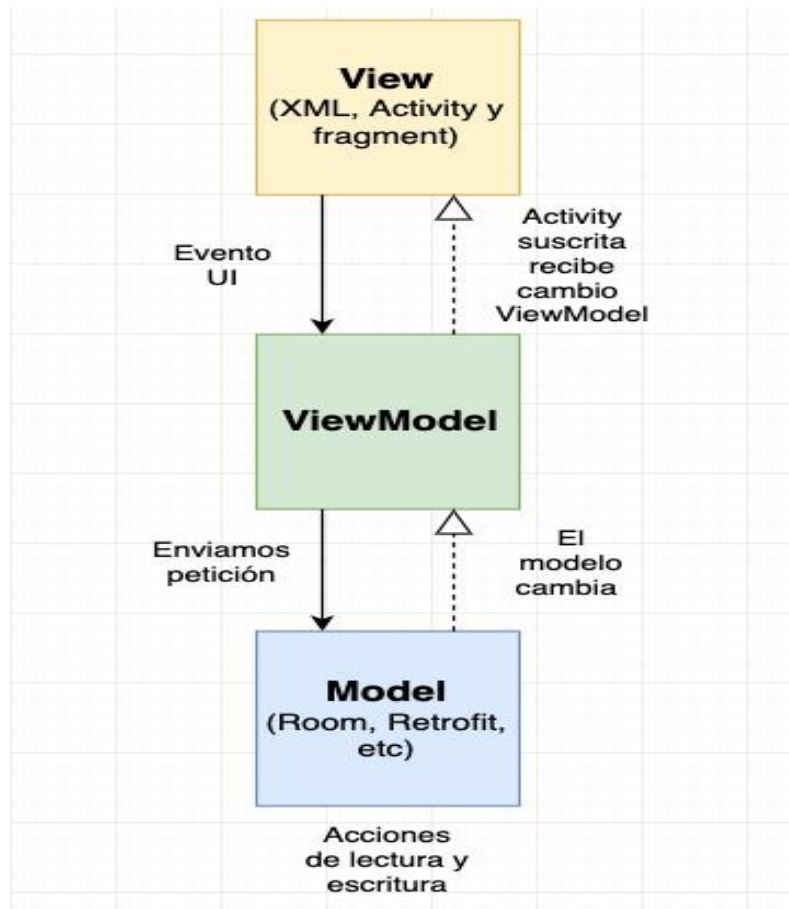


Figura 2. Diseño arquitectónico: Patrón Modelo Vista - Vista Modelo

Fuente: (Reyes, 2018)

En la siguiente figura se describe el diseño arquitectónico de la propuesta de solución

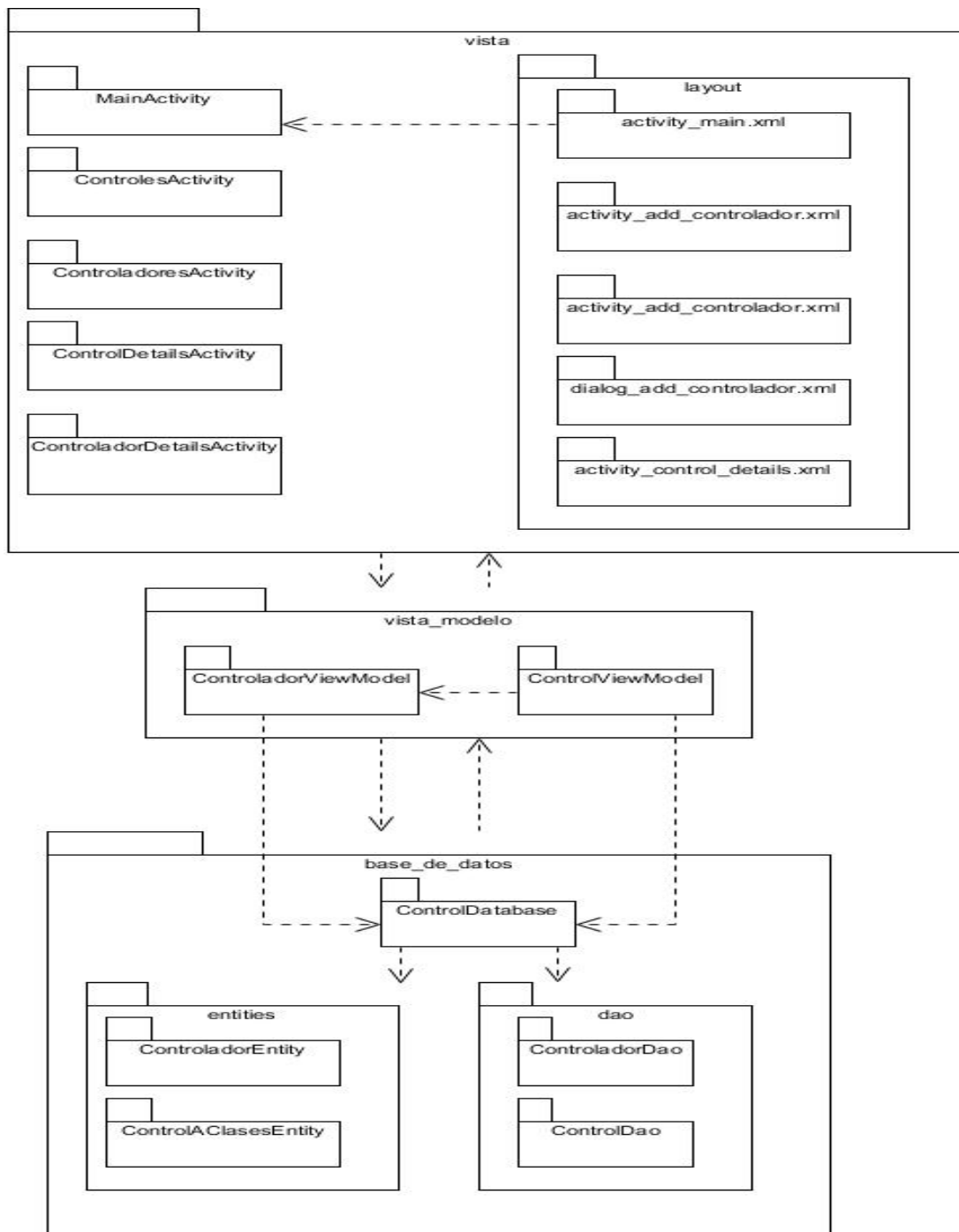


Figura 3. Diseño arquitectónico de la propuesta de solución

Fuente: Elaboración propia

En el diseño anterior se puede evidenciar la estructura arquitectónica de la propuesta de solución basado en el patrón MVVM, las diferentes clases que componen la aplicación para la gestión de información de los controles a clases se organizan en las tres capas siguientes

- **View:** Contiene las vistas *MainActivity*, *ControlesActivity*, *ControladoresActivity*, *ControlDetailsActivity*, *ControladoresDetailsActivity* y las plantillas que corresponden a cada una de las vistas de la solución: *activity_main.xml*, *activity_add_controlador.xml*, *dialog_add_controlador.xml*, *activity_control_details.xml*.
- **ViewModel:** Contiene los controladores para acceder al modelo e interactuar con las vistas correspondientes: *ControladorViewModel* y *ControlViewModel*.
- **Database:** Contiene las clases entidades *ControladorEntity* y *ControlAClasesEntity* las clases de acceso a datos *ControladorDao* y *ControlDao* y la clase que gestiona la conexión con la base de datos *ControlDatabase*.

2.3.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. El mismo identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades (Martínez, 2020).

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP del inglés *General Responsibility Assignment Software Patterns*) tienen una importante utilidad en el diseño de una aplicación, al igual que los *Gang-of-Four* o Pandilla de los Cuatro (*GoF* por sus siglas en inglés). A continuación, se muestra una selección de estos patrones los cuales serán utilizados durante el diseño de la aplicación:

Experto

Este patrón tiene como objetivo principal asignar una responsabilidad determinada a la clase que tenga la mayor cantidad de información para hacer esta tarea. Es la razón anterior la que le da su apellido Experto “en información” (Bustacara, 2017). Este patrón se evidencia en la clase *ControlViewModel* es quien tiene acceso a todas las entidades necesarias y por ello a la información que en ellas se encuentra.

Bajo Acoplamiento

El acoplamiento es una medida de la fuerza en que una clase está conectada a otras, que la conoce y recurre a ellas. El objetivo de este patrón consiste en mantener un bajo nivel de dependencia de otros elementos, por lo que constituye un principio que debe estar presente en todas las decisiones de diseño

con lo que se reduce el impacto de los cambios (Bustacara, 2017). Este patrón se evidencia en la clase *ControlesActivity*.

Alta Cohesión

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Se puede afirmar que cada una de las clases del sistema tiene alta cohesión, de manera que estas poseen la característica de tener las responsabilidades estrechamente relacionadas. Esta particularidad evita en cada caso, tener que realizar un trabajo enorme al garantizar un mejor diseño en ocasiones para el resultado global (Bustacara, 2017). Este patrón se evidencia en la clase *ControlEntity*.

2.3.3 Diseño de clases

El modelo de diseño se utiliza como medio de abstracción del modelo de implementación y el código fuente del software. Su objetivo fundamental es transmitir, a través de la representación mediante diagramas, una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones concernientes a los lenguajes de programación (Creately, 2022).

En la siguiente figura se muestra un ejemplo de Diagramas de Clases de Diseño pertenecientes a la aplicación para el control docente a profesores.

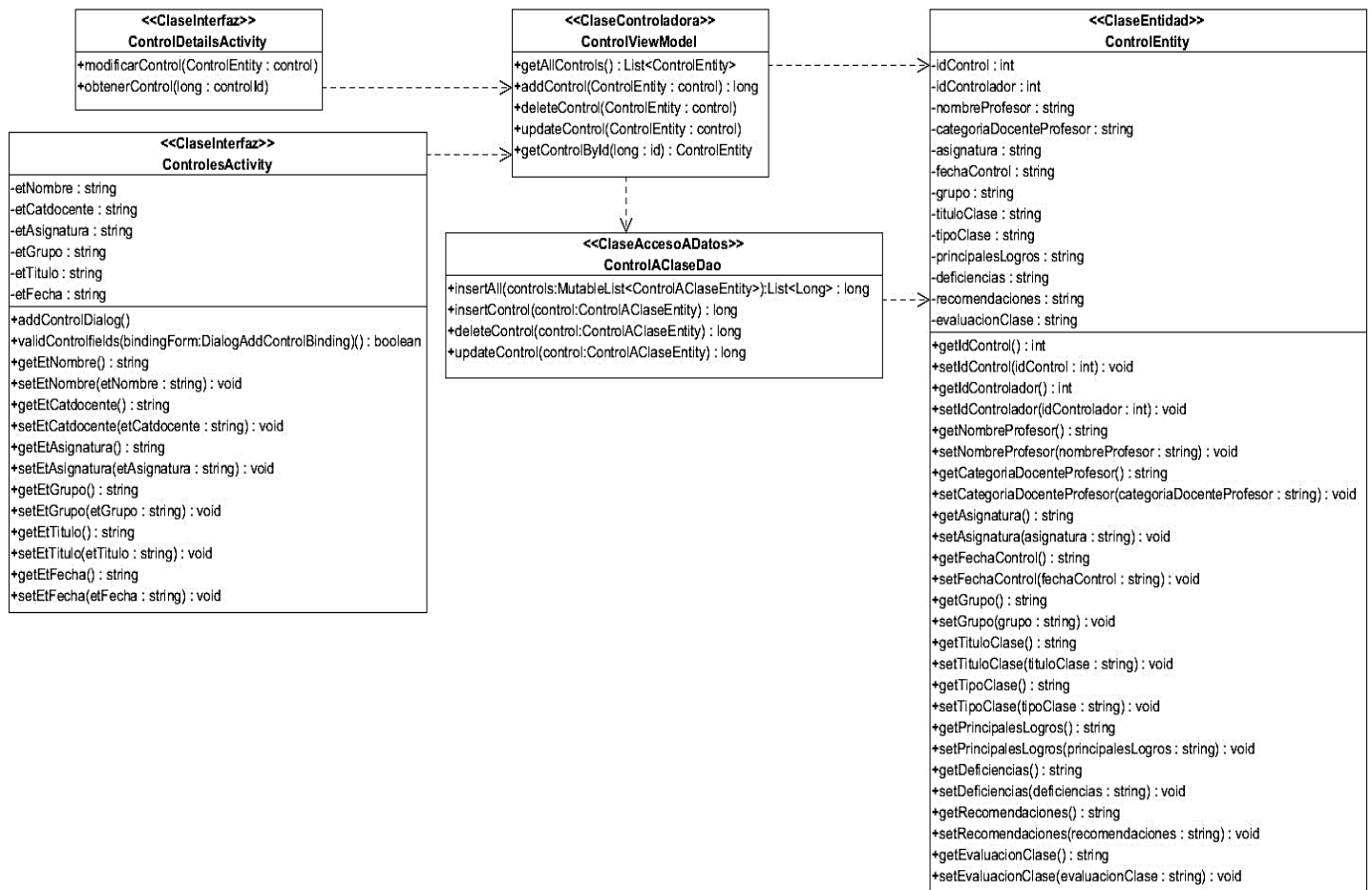


Figura 4. Diagrama de clases de la propuesta de solución Diagrama de clases

Fuente: Elaboración propia

El diagrama de clases propuesto anterior está formado por las siguientes clases.

- Clases interfaces: permiten la comunicación de los usuarios con la aplicación. Comprende las vistas *ControlDetailActivity* y *ControlesActivity*.
- Clase controladora: contiene las funcionalidades necesarias para interactuar con las clases interfaces, las clases entidades y las clases de acceso a datos. Este caso es la clase *ControlViewModel*.
- Clase de acceso a datos: posibilita la comunicación entre la clase controladora y la base de datos. Está representada por la clase *ControlAclaseDao*.
- Clase entidad: representa el almacenamiento de los datos persistentes en el tiempo, incluso cuando haya finalizado la ejecución de la aplicación. Está formada por la clase *ControlEntity*.

2.3.3 Diseño de datos

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos o podríamos decir que es un conjunto de conceptos que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos (Redondo, 2017).

Principales características de un modelo de datos:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Respaldo y recuperación.
- Seguridad de acceso y auditoría.
- Integridad de los datos.
- Consultas complejas optimizadas.

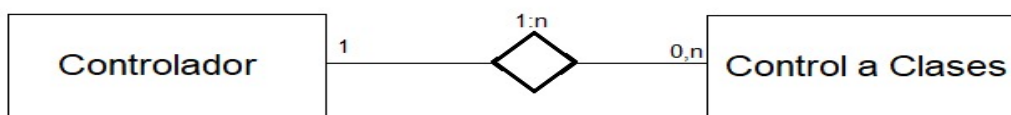


Figura 5. Diagrama de Entidad Relación de Base de Datos

Fuente: Elaboración propia

2.4 Conclusiones del capítulo

Con la especificación de los requisitos funcionales y no funcionales del sistema, se logró una mejor comprensión de los resultados que se pretenden obtener de una manera precisa y sirven de guía para la implementación del sistema. La representación y descripción de los artefactos generados garantizaron un mejor entendimiento de los flujos de trabajos presentes en el proceso de generar el reporte estadístico. La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del sistema web propuesto, lo que garantizó una mayor organización e hizo el código más legible. La elaboración del diagrama de despliegue permitió identificar la disposición física de los artefactos de la aplicación para el control docente a profesores.

CAPÍTULO 3: Implementación y evaluación de la aplicación para dispositivos móviles que permite mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4

Introducción

En el presente capítulo se describen los diferentes mecanismos utilizados para llevar a cabo el desarrollo y validación del sistema propuesto. Así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software para un mejor entendimiento y organización del mismo. Luego se realiza la fase de prueba, cuyo objetivo es comprobar si el sistema cumple con los requerimientos.

3.1 Implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (Rodríguez, 2015).

3.1.1 Estándares de codificación utilizados

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. El estándar de codificación debería establecer cómo operar con la base de código existente (Hicken, 2020). Seguidamente se especifican los estándares de codificación a utilizar en la construcción de la solución:

- El tamaño máximo de las líneas de código debe ser de cien a ciento veinte caracteres aproximadamente, de manera tal que se garantice la completa visibilidad de las líneas de código sin necesidad de realizar desplazamiento horizontal.

```
private fun initRecyclerView() {  
    binding.rvControllist.layoutManager = LinearLayoutManager(this)  
    controlAdapter = ControlAdapter(controllist, controlViewModel)  
    binding.rvControllist.adapter = controlAdapter  
}
```

Figura 6. Ejemplo de empleo de estándar de codificación: tamaño máximo de las líneas de código

Fuente: Elaboración propia

- Los nombres de las clases y las funciones adoptarán la notación *CamelCase* y no se utilizará el guion bajo como delimitador entre palabras.

```
private fun validControlFields(bindingForm: DialogAddControlBinding): Boolean {
```

Figura 7. Ejemplo de empleo de estándar de codificación: nombres de las clases y las funciones

Fuente: Elaboración propia

- **Número de declaraciones por línea:** se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

```
val catDoc = bindingForm.etCatDocente.text.toString()
val asig = bindingForm.etAsignatura.text.toString()
val grupo = bindingForm.etGrupo.text.toString()
val titulo = bindingForm.etTitulo.text.toString()
val fecha = bindingForm.etFecha.text.toString()
```

Figura 8. Ejemplo de empleo de estándar de codificación: declaración de variables

Fuente: Elaboración propia

➤ Sentencias

Las sentencias pertenecientes a un bloque de código estarán ubicadas a un nivel más a la derecha con respecto a la sentencia que las contiene. El carácter inicio de bloque "{" debe situarse al final de la línea que inicia el bloque y "}" debe situarse en una nueva línea tras la última línea del bloque y alineada con respecto al primer carácter de dicho bloque. Todas las sentencias de un bloque deben encerrarse entre llaves "{...}", aunque el bloque contenga solo una sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves.

```
private fun validControlFields(bindingForm: DialogAddControlBinding): Boolean {
    return !(bindingForm.etNombre.text.isEmpty() ||
            bindingForm.etCatDocente.text.isEmpty() ||
            bindingForm.etAsignatura.text.isEmpty() ||
            bindingForm.etGrupo.text.isEmpty() ||
            bindingForm.etFecha.text.isEmpty() ||
            bindingForm.etTitulo.text.isEmpty())
}
```

Figura 9. Ejemplo de empleo de estándar de codificación: niveles de codificación

Fuente: Elaboración propia

3.2 Evaluación de la aplicación desarrollada

En la ingeniería de software la validación es el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido, se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.

3.3 Pruebas de software

Las pruebas de software comprenden una fase del proceso de desarrollo que se centra en asegurar la calidad, fiabilidad y robustez de un software, dentro de un contexto o escenario donde está previsto que este sea utilizado. Se encuentra encaminado a medir el cumplimiento de las funcionalidades establecidas por el cliente, reduciendo de esta manera el número de errores no detectados. Esta fase es una de las más costosas del ciclo de vida del software. En sentido estricto, deben realizarse pruebas a todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación (Loadview, 2020).

La estrategia a seguir para la realización de las pruebas a la aplicación móvil implementada, contemplan dos niveles de pruebas: pruebas Unitarias y de Aceptación.

3.3.1 Pruebas Unitarias

Las pruebas unitarias se realizan para controlar el funcionamiento de pequeñas porciones de código como métodos o clases. Generalmente son realizadas por los mismos programadores puesto que al conocer con mayor detalle el código, se les simplifica la tarea de elaborar conjuntos de datos de prueba para testarlo. Por último, es importante que las funcionalidades de cada componente unitario sea cubierta, por al menos, dos casos de prueba, los cuales deben centrarse en probar al menos una funcionalidad positiva y una negativa (Moreno, 2019).

Las pruebas de unidad siempre están orientadas a caja blanca y le permiten al programador conocer si determinada funcionalidad se puede agregar al sistema sin afectar su funcionamiento. Facilitan que el programador cambie el código para mejorar su estructura (refactorización), puesto que permiten hacer pruebas sobre los cambios y asegurarse de que no han introducido errores.

Para la realización de las pruebas unitarias es utilizada la extensión de Android *JUnit*, que es una parte integral del SDK de Android. A cada una de las clases de la aplicación se le realizaron varias pruebas con el objetivo de obtener y corregir los errores existentes antes de realizar la entrega al cliente.

Resultado de las pruebas unitarias

Para probar el correcto funcionamiento de la aplicación móvil se realizaron tres iteraciones de pruebas. En la tabla que a continuación se presenta, se muestran los resultados obtenidos en cada iteración de prueba a la aplicación móvil para la gestión de controles a clases, así como la corrección de cada uno de los errores.

Tabla 5. Cantidad de no conformidad por cada iteración

Fuente: Elaboración propia

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	3	2	0
Resueltas	3	2	0
Pendientes	0	0	0

En la tabla 5 se muestra el comportamiento de las no conformidades encontradas durante el proceso de prueba donde se observa que en la primera iteración se encontraron tres (3) no conformidades de validaciones incorrectas, en la segunda iteración se encontraron dos (2) no conformidades, de ellas una (1) al cargar la vista de detalles de un control a clases no carga los datos del control, una (1) la funcionalidad de adicionar controles a clases de error al pulsar el botón de adicionar.

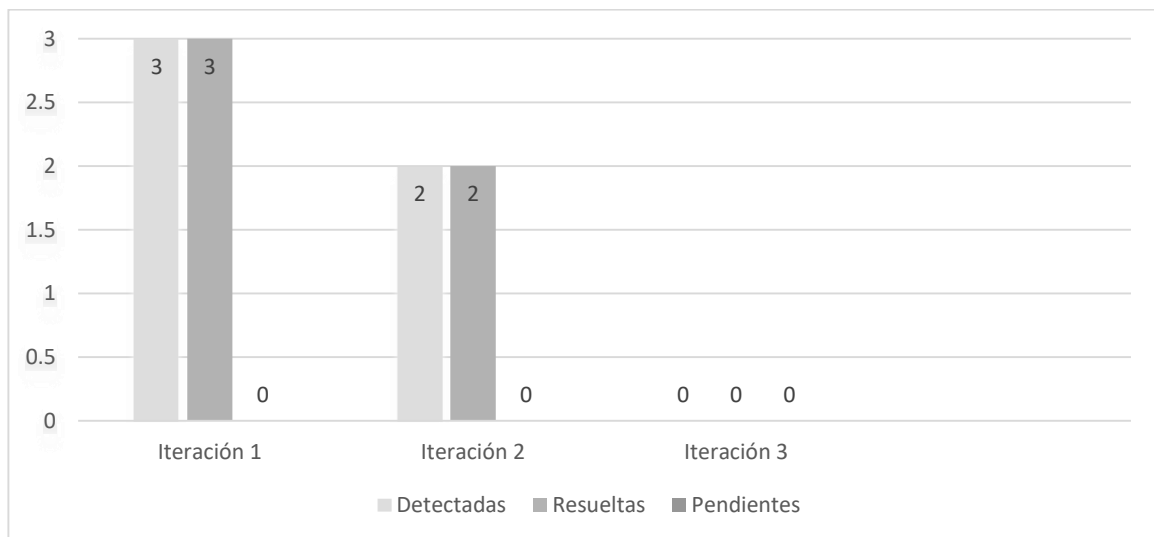


Figura 10. Comportamiento de las no conformidades en las pruebas unitarias

Fuente: Elaboración propia

3.3.2 Prueba de Aceptación

Las pruebas de aceptación son básicamente pruebas funcionales sobre el sistema completo, ya que tienen como objetivo obtener la aceptación final del cliente antes de la entrega del producto para su utilización. Son realizadas por el usuario final permitiendo la valoración del producto, donde el cliente confirma que las funcionalidades exigidas y descritas en la HU funcionan correctamente. Intentan encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos, errores de rendimiento y errores de inicialización y terminación (Digite, 2022).

Para representar las pruebas de aceptación se definieron los siguientes elementos:

- **CU:** Número de la UH a la cual pertenece.
- **Nombre:** Junto al código conforma el identificador del caso de prueba.
- **Descripción:** Acción que debe realizar el sistema.
- **Condiciones de ejecución:** Describe las características y elementos que debe contener el sistema para realizar el caso de prueba.
- **Entrada/Pasos de Ejecución:** Incluye las entradas necesarias para realizar el sistema, además de los pasos para realizar el caso de prueba.
- **Resultados Esperados:** Descripción de la respuesta del sistema ante el caso de prueba.
- **Evaluación de la prueba:** Clasificación de la prueba en satisfactoria o insatisfactoria.

La siguiente tabla muestra las pruebas de aceptación del CU1 perteneciente a la primera iteración de sistema. La cual fue escogida por ser una de los procesos más relevantes en el desarrollo del sistema.

Tabla 6. Caso de prueba insertar control a clases

Fuente: Elaboración propia

Caso de Prueba Aceptación
CU: HU_1
Nombre: Insertar control a clases
Nombre de la persona que realiza la prueba: Yadira Rodríguez Ramírez
Descripción de la Prueba: La aplicación permite al usuario crear un control a clase. En caso de haber dejado alguno de los campos vacíos se enviara un mensaje notificando al usuario que existen campos por completar “Existen campos vacíos”.
Condiciones de Ejecución: El usuario selecciona la opción de “Adicionar control a clases” del menú principal. Luego aparecen los campos para insertar nombre de la persona a la que se le va a realizar el control, grupo, título de la clase, nombre del controlador. Seleccionar la categoría docente del controlado, categoría docente del controlador, asignatura y tipo de clase. Una vez llenados los campos se selecciona la opción de “Adicionar” en el menú inferior.
Entradas/ Pasos de Ejecución: Seleccionar en el botón insertar, a continuación aparecen los campos a llenar “Nombre y Apellidos, Categoría Docente, Asignatura, Grupo, Titulo de la Clase, Control Realizado por, Categoría Docente, Tipo de Clase”.
Resultado esperado: El usuario insertó el control a clases y se notifica el éxito de la creación del control a clases.
Evaluación de la prueba: Prueba satisfactoria

Tabla 7. Caso de prueba eliminar control a clases

Fuente: Elaboración propia

Caso de Prueba Aceptación
CU: HU_2
Nombre: Eliminar control a clases
Nombre de la persona que realiza la prueba: Yadira Rodríguez Ramírez
Descripción de la Prueba: La aplicación permite al usuario eliminar un control a clase. A continuación se muestra un mensaje de confirmación ¿Está seguro que desea eliminar el elemento? Y en la parte de inferior se muestran los botones de cancelar y aceptar que deberá seleccionar para confirmar su petición.
Condiciones de Ejecución: El usuario va al listado de controles realizados y selecciona el control a clases que desea eliminar.
Entradas/ Pasos de Ejecución: El usuario va al listado de controles realizados y selecciona el control a clases que desea eliminar, lo deja presionado y a continuación se muestra un mensaje de confirmación.
Resultado esperado: El usuario eliminó el control a clases y se notifica el éxito de la eliminación del control a clases.
Evaluación de la prueba: Prueba satisfactoria

Tabla 8. Cantidad de no conformidades por cada iteración

Fuente: Elaboración propia

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	3	1	0
Resueltas	3	1	0
Pendientes	0	0	0

En la tabla 8 se muestra el comportamiento de las no conformidades encontradas durante el proceso de prueba donde se observa que en la primera iteración se encontraron tres (3) no conformidades de errores

ortográficos, en la segunda iteración se encontró una (1) no conformidad de presentación porque no se visualiza correctamente toda la información que se desea mostrar.

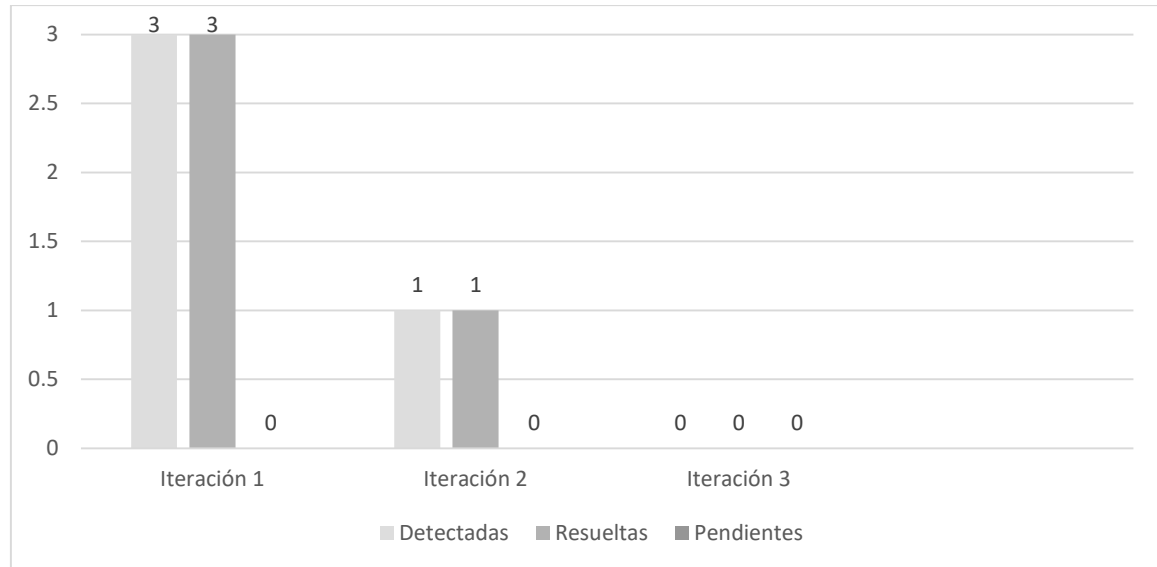


Figura 11. Comportamiento de las no conformidades en las pruebas de aceptación

Fuente: Elaboración propia

3.4 Conclusiones del capítulo

La implementación de la propuesta de solución permitió mejorar el proceso de gestión de información sobre el control a clases en la Facultad 4, mediante una aplicación para dispositivos móviles. La utilización de estándares de codificación posibilitó adoptar una estructura homogénea que facilita el entendimiento del software desarrollado, así como su mantenimiento y una menor cantidad de errores; logrando un código más limpio y fácil de mantener. Las pruebas realizadas permitieron detectar los errores presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con mayor calidad.

CONCLUSIONES GENERALES

Luego de finalizada la investigación, se pudo arribar a las siguientes conclusiones:

- La elaboración y análisis del marco teórico sobre el proceso de gestión de información del control a clases, así como el estudio de sistemas homólogos demostraron la necesidad de implementar una aplicación que resolviera las deficiencias planteadas al inicio de la investigación. Para ello se emplearon tecnologías que garantizan la soberanía tecnológica del país siguiendo un proceso de desarrollo de software guiado por la metodología AUP para la UCI.
- La identificación de los 7 requisitos funcionales garantizó que la propuesta de solución respondiera a las necesidades planteadas por el cliente, quedando explícito lo que la aplicación debía hacer y con los 7 requisitos no funcionales se evidenció claramente las cualidades del producto. La modelación de los artefactos facilitó el soporte a la implementación de los requisitos previamente expresados por el cliente y garantizó la base para la organización lógica del código fuente.
- La solución desarrollada permitió la obtención de una aplicación móvil para la gestión de controles a clases, disminuyendo el tiempo necesario para su ejecución y evitar las pérdidas de informes de clases.
- La realización de las pruebas de software erradicó las insuficiencias detectadas en la aplicación, permitiendo su corrección en aras de lograr un producto con calidad y que cumpliera las necesidades del cliente.

RECOMENDACIONES

Una vez concluida la investigación y el desarrollo de la propuesta de solución se recomienda:

- Integrar la aplicación móvil al sistema de gestión académica, Akademos.

REFERENCIAS BIBLIOGRÁFICAS

ÁLVAREZ, C. *¿Qué es Gradle?* [En línea] *¿Qué es Gradle?*, 2015. [Citado el: 22 de mayo de 2022].

Disponible en: <https://www.arquitecturajava.com/que-es-gradle/>

ARAQUE, W. *La vinculación con la colectividad y la articulación universidad-sociedad.* [En línea] *La vinculación con la colectividad y la articulación universidad-sociedad*, 2012. [Citado el: 20 de abril de 2022].

Disponible en: <https://www.redalyc.org/pdf/834/83449754002.pdf>

ARIAS, A. *Especificación de requisitos de software.* [En línea] *Especificación de requisitos de software*, 2019. [Citado el: 3 de julio de 2022]. Disponible en:

[https://repository.ucc.edu.co/bitstream/20.500.12494/10954/1/%282019%29_Especificaci%C3%B3n de R equisitos de Software Petic.pdf](https://repository.ucc.edu.co/bitstream/20.500.12494/10954/1/%282019%29_Especificaci%C3%B3n_de_Requisitos_de_Software_Petic.pdf)

BERMUDEZ, R.; RODRÍGUEZ, M. *Teoría y metodología del aprendizaje.* [En línea] *Teoría y metodología del aprendizaje*, 1996. [Citado el: 15 de abril de 2022]. Disponible en:

<https://www.redalyc.org/pdf/834/83449754002.pdf>

BERNAL, D. *Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos.* [En línea] *Principales tipos de aplicaciones móviles: ventajas, desventajas y ejemplos*, 2021. [Citado el 20 de julio de 2022]. Disponible en:

<https://profile.es/blog/tipos-aplicaciones-moviles-ventajas-ejemplos/>

BESCÓS, A. *Desarrollo de software importancia y técnicas de la captura de requisitos.* [En línea] *Desarrollo de software importancia y técnicas de la captura de requisitos*, 2019. [Citado 25 de julio de 2022]. Disponible en:

<https://www.atsistemas.com/es/blog/desarrollo-de-software-importancia-y-tcnicas-de-la-captura-de-requisitos>

BUSTACARA, C.; DÍAZ, C. L., et al. *Análisis y diseño O.O. Arquitectura de software.* [En línea] *Análisis y diseño O.O. Arquitectura de software*, 2007. [Citado el: 23 de mayo de 2022]. Disponible en:

https://sophia.javeriana.edu.co/~lcdiaz/ADOO2006-3/grasp_cpaternostro-lvargas-iviafara.pdf.

COMPARASOFTWARE. *Tabalive Gestión de Aula.* [En línea] *Tabalive Gestión de Aula*, 2021. [Citado el 20 de mayo de 2022]. Disponible en: <https://www.comparasoftware.com/tabalive-gestion-de-aula>

CONTROL-L. *¿Cómo funciona Control L?* [En línea] *¿Cómo funciona Control L?*, 2016. [Citado el: 18 de mayo de 2022]. Disponible en: https://www.control-l.com/como_funciona.asp

CREATELY. *La Guía Fácil de los Diagramas de Clases UML | Tutorial de Diagramas de Clases* [En línea] La Guía Fácil de los Diagramas de Clases UML | Tutorial de Diagramas de Clases, 2022. [Citado el 5 de agosto de 2022]. Disponible en: <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-clases/>

DALVIK, JOSÉ. *El Conspirador. Qué es y para qué sirve un modelo conceptual.* [En línea] El Conspirador. Qué es y para qué sirve un modelo conceptual, 2018. [Citado el: 10 de febrero de 2022]. Disponible en: <https://www.elconspirador.com/2013/12/21/que-es-y-para-quesirve-un-modelo-conceptual/>

DELGADO, C. *¿Qué es IDE (entorno de desarrollo integrado)?* [En línea]. *¿Qué es IDE (entorno de desarrollo integrado)?*, 2021. [Citado el 15 de mayo de 2022]. Disponible en: <https://ourcodeworld.co/articulos/leer/1469/que-es-un-ide-entorno-de-desarrollo-integrado>

DIGITÉ. *Pruebas de aceptación el qué y el por qué.* [En línea] Pruebas de aceptación el qué y el por qué, 2022. [Citado el: 5 de agosto de 2022]. Disponible en: <https://www.digite.com/es/agile/pruebas-de-aceptacion/>

DOMAIN. *Metodología del desarrollo de software.* [En línea] Metodología del desarrollo de software, 2021. [Citado el 22 de julio de 2022]. Disponible en: <https://domainlogic.io/metodologias-de-desarrollo-de-software-2022/>

EDUCACIÓN 3.0. *Additio App.* [En línea] Additio App, 2022. [Citado el 20 de mayo de 2022]. Disponible en: <https://www.educacionrespuntocero.com/recursos/plataformas-gestion-escolar/>

GARCIA, A. *¿Qué es Maven y para qué se utiliza?* [En línea] *¿Qué es Maven y para qué se utiliza?*, 2015. [Citado el: 22 de mayo de 2022]. Disponible en: <http://panamahitek.com/que-es-maven-y-para-que-se-utiliza/>

GAVILÁN, I. *El contexto de los procesos de negocio.* [En línea] El contexto de los procesos de negocio. [Citado el 5 de mayo de 2022]. Disponible en: <https://ignaciogavilan.com/el-contexto-de-los-procesos-de-negocio/>

GÓMEZ, I. *¿Qué es GitLab?* [En línea] *¿Qué es GitLab?*, 2022. [Citado el: 22 de mayo de 2022]. Disponible en: <https://www.crehana.com/blog/desarrollo-web/que-es-gitlab/>

GUERRA, A. *Software Guru. Obtención de requerimientos. Técnicas y Estrategias.* [En línea] Software Guru. Obtención de requerimientos. Técnicas y Estrategias, 2018. [Citado el: 28 de abril de 2022]. Disponible en: <https://sg.com.mx/revista/17/obtencionrequerimientos-tecnicas-y-estrategia>

GUIMERÁ, A. *¿Qué es Kotlin y características?* [En línea] ¿Qué es Kotlin y características?, 2021.

[Citado el: 22 de mayo de 2022]. Disponible en: <https://openwebinars.net/blog/que-es-kotlin/>

GUZMÁN, C. *El proceso extensionista universitario como vía para la pertinencia en la formación del futuro profesional.* [En línea] El proceso extensionista universitario como vía para la pertinencia en la formación del futuro profesional, 2014.

[Citado el 10 de mayo de 2022]. Disponible en: <https://www.redalyc.org/pdf/834/83449754002.pdf>

HERNÁNDEZ I. *La clase en la educación superior, forma organizativa esencial en el proceso de enseñanza-aprendizaje.* [En línea] La clase en la educación superior, forma organizativa esencial en el proceso de enseñanza-aprendizaje, 2017.

[Citado el: 10 de mayo de 2022]. Disponible en: <https://www.redalyc.org/articulo.oa?id=83449754002>

HIKEN A. *Una onza de prevención: seguridad y protección a través de estándares de codificación de software.* [En línea] Una onza de prevención: seguridad y protección a través de estándares de codificación de software, 2020.

[Citado el 6 de julio de 2022]. Disponible en: <https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>

IFT. *Telefonía móvil.* [En línea] Telefonía móvil, 2015. [Citado el: 15 de mayo de 2022]. Disponible en:

<http://www.ift.org.mx/usuarios-telefonía-movil/sabias-que-la-telefonía-movil>

LOADVIEW. *Tipos de pruebas de software.* [En línea] Tipos de pruebas de software. [Citado el: 5 de julio de 2022].

Disponible en: <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>

LUCIDCHART. *Como diseñar una arquitectura de software.* [En línea] Como diseñar una arquitectura de software, 2022.

[Citado el 25 de julio de 2022]. Disponible en: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software>

MARTÍN, G. *Definición de requisitos funcionales bajo especificación IEEE para un sistema de ingeniería.*

[En línea] Definición de requisitos funcionales bajo especificación IEEE para un sistema de ingeniería.

[Citado el 1 de julio de 2022]. Disponible:

https://oa.upm.es/56018/1/TFG_GONZALO_MARTIN_DE_LAS_PUEBLAS_ENCINAS.pdf

MARTÍNEZ, M. *¿Qué son los patrones de diseño de software?* [En línea] ¿Qué son los patrones de diseño de software?

[Citado el 16 de julio de 2022]. Disponible: <https://profile.es/blog/patrones-de-diseno-de-software/>

MENDEZ, R. *Principales herramientas CASE del mercado y su uso.* [En línea] Principales herramientas CASE del mercado y su uso, 2020. [Citado el 4 de julio de 2022]. Disponible en: https://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html

MES. *La Nueva Universidad Cubana y su contribución a la universalización del conocimiento.* [En línea] La Nueva Universidad Cubana y su contribución a la universalización del conocimiento, 2006. [Citado el: 5 de mayo de 2022]. Disponible en: <https://www.redalyc.org/pdf/834/83449754002.pdf>

MES. *Resolución No. 2/2018 (GOC-2018-O25) Reglamento del trabajo docente metodológico en la educación superior.* [En Línea] Resolución No. 2/2018 (GOC-2018-O25) Reglamento del trabajo docente metodológico en la educación superior, 2018. [Citado el: 7 de mayo de 2022]. Disponible en: <https://www.redalyc.org/pdf/834/83449754002.pdf>

MORENO O. *Pruebas unitarias.* [En línea] Pruebas unitarias, 2019. [Citado el 4 de julio de 2022]. Disponible en: <http://oscarmoreno.com/pruebas-unitarias/>

NOBBOT. *¿Qué es Android Studio?.* [En línea] ¿Qué es Android Studio?, 2021. [Citado el: 21 de mayo de 2022]. Disponible en: <https://www.nobbot.com/tecnologia/aplicaciones-moviles-tecnologia/android-studio-que-es-como-descargarlo/>

OBJECT MANAGEMENT GROUP. *Introduction to OMG's Unified Modeling Language.* [En línea] Introduction to OMG's Unified Modeling Language, 2015. [Citado el: 23 de Noviembre de 2016]. Disponible en: http://www.omg.org/gettingstarted/what_is_uml.htm

OROCIO, M. *¿Qué es un modelo de implementación?.* [En línea] ¿Qué es un modelo de implementación?, 2022. [Citado el: 25 de julio de 2022]. Disponible en: <https://www.eluniversadomex.mx/que-es-un-modelo-de-implementacion>

MOINFORMATICA. *Requerimientos no funcionales* [En línea] Requerimientos no funcionales, 2015. [Citado el: 23 de mayo de 2022]. Disponible en: <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>

REDONDO, M. *Estudio con modelos de datos para la automatización en redes eléctricas inteligentes.* [En línea]. Estudio con modelos de datos para la automatización en redes eléctricas inteligentes. [Citado el: 31 de mayo del 2017]. Disponible en: <http://hdl.handle.net/10396/14506>

RODRÍGUEZ, T. *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana, 2015

TURNEMANN, C.; TOBÓN, S. *La educación superior en el umbral del siglo XXI*. [En línea] La educación superior en el umbral del siglo XXI, 2006. [Citado el: 1 de mayo de 2022]. Disponible en: <https://www.redalyc.org/pdf/834/83449754002.pdf>

VERGARA, S. *¿Qué son las historias de usuario?* [En línea] ¿Qué son las historias de usuario?, 2021. [Citado el: 23 de mayo de 2022]. Disponible en: <https://www.itdo.com/blog/que-son-las-historias-de-usuario/>

VISUAL PARADIGM INTERNATIONALS. *Visual Paradigm*. [En línea] Visual Paradigm, 2018. [Citado el: 21 de mayo de 2022]. Disponible en: <http://www.visual-paradigm.com/>.

ZAMBRANO, R. A. *Metodología de la investigación*. [En línea] Metodología de la investigación, 2013. [Citado el: 21 de mayo de 2022]. Disponible en: <http://es.slideshare.net/Mal3J1Ta/resumen-capitulos-1234-del-libro>

ANEXOS

Anexo 1: Entrevista realizada a la Jefa del Departamento de Informática de la Facultad 4

Objetivo: conocer cómo se realiza el proceso de control a clases en la facultad, qué información se procesa sobre estos controles y qué deficiencias persisten en la actualidad en la ejecución de dicho proceso.

1. ¿Considera usted necesario el desarrollo de controles a las clases que imparten los profesores en la facultad? ¿Por qué?
2. ¿Qué características tiene este proceso?
3. ¿Qué información se obtiene de la ejecución de un control a clases?
4. ¿Cómo se procesa dicha información?
5. ¿Existen deficiencias en el desarrollo de los controles a las clases y la forma en la que se manipula la información obtenida? ¿Cuáles?
6. Consideras que el desarrollo de una aplicación para dispositivos móviles contribuye a mejorar el proceso de controles a las clases impartidas por los docentes de la facultad.

Anexo 2: Guía de observación para analizar aplicaciones informáticas que gestionan información sobre el control a clases

Observadora: Marlin Remigio Díaz

Objetivo: Identificar los elementos fundamentales sobre aplicaciones informáticas que gestionan información sobre el control a clases.

Elementos a observar:

1. Datos de las aplicaciones informáticas:

- Nombre de las aplicaciones
- Tipo de licencia
- Cuenta con interfaz gráfica de usuario
- Compatibilidad con diferentes sistemas operativos

2. Características de las aplicaciones informáticas:

- ¿Cómo están estructuradas las interfaces de las aplicaciones?
- ¿Qué información se procesa en ellas?
- ¿Qué funcionalidades le brinda al usuario?
- ¿Qué herramientas necesitan para ser instaladas?
- ¿Cuáles son los pasos a seguir para instalar estas herramientas?

3. Impacto social de las aplicaciones informáticas.

Anexo 3: Encuesta aplicada a la jefa del Departamento de Informática de la facultad 4

Objetivo: identificar funcionalidades y características que debe poseer la propuesta de solución.

1. ¿Cuáles son los pasos se siguen en el control a clases?
2. ¿Qué funcionalidades consideras que debe tener la aplicación para dispositivos móviles?
3. ¿En qué formato se debe exportar la información procesada por la aplicación?
4. ¿Qué características deben tener las interfaces gráficas de usuario de la aplicación?

Anexo 4: Planilla del Informe de control a clases

INFORME DE CONTROL A CLASE

Nombres y Apellidos del docente controlado:

Categoría Docente:

Asignatura: _____ Grupo: _____

Título de la clase:

Control realizado por: _____ Categoría Docente: _____

Tipo de clase: Conferencia ___ Laboratorio ___ Clase Práctica ___ Práctica Laboral ___ Seminario ___

|

No.	Indicadores
	I. Aspectos organizativos de la clase
1.1	Correspondencia con el Programa y el Plan Calendario de la asignatura (P1)
1.2	Comienzo de la clase a la hora prevista, asistencia y puntualidad de los estudiantes y el profesor.
	II. Tratamiento didáctico-metodológico
2.1	Orientación hacia el cumplimiento del objetivo de la actividad.
2.2	Organización, tratamiento uso de los métodos de enseñanza y dominio de los contenidos: a) Aseguramiento del nivel de partida de los estudiantes, su auto-preparación y motivación para la actividad. b) Ambiente que propicia durante el proceso de enseñanza-aprendizaje (orden, atención, interés y participación activa de los estudiantes, trabajo cooperativo). c) Actividad Centrada en el aprendizaje del estudiante, mediante el empleo de métodos activos y del papel orientador del profesor. d) Realización de conclusiones parciales y finales.
2.3	Uso de formas de control y/o evaluación del aprendizaje.
2.4	Orientación del estudio trabajo independiente, con indicadores claros y medibles para su ejecución y control.
2.5	Uso de los medios de enseñanza aprendizaje.
	III. Aplicación de las estrategias curriculares
3.1	Comunicación profesor-estudiante y estudiante-estudiante. Formación en valores de los estudiantes a través de los contenidos de la clase.
3.2	Cumplimiento de la estrategia de Trabajo Político-Ideológico desde la clase.
3.3	Cuidado y rigor en la exposición de sus ideas, precisión en el lenguaje oral y escrito.
3.4	Vinculación con otras materias del año o de la carrera.
	IV. Cumplimiento de los objetivos de la clase
4.1	Se cumplieron los objetivos definidos para la clase.

Principales logros:

Deficiencias:

Recomendaciones:

Evaluación de la clase:

Fecha del control:

Firma del controlador

Firma del controlador

Firma del profesor visitado

Figura 12. Informe de Control a Clases

Fuente: Elaboración propia del autor

Anexo 5: Historias de Usuarios

Tabla 9. HU_3: Modificar control a clases

Fuente: Elaboración propia del autor

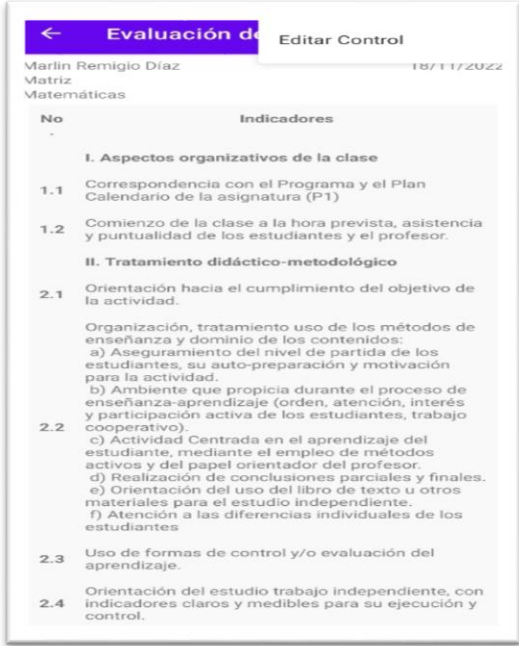
Historia de usuario	
Número: HU_3	Nombre: Modificar control a clases.
Prioridad en negocio: Media	
<p>Descripción: Permite al usuario modificar un control a clases. El usuario va al listado de controles realizados y selecciona el control a clases que desea modificar y lo deja presionado y a continuación se muestra un mensaje de confirmación ¿Está seguro que desea eliminar el elemento?, luego en la parte inferior se muestran los botones de cancelar y aceptar q deberá seleccionar para confirmar su petición.</p>	
<p>Prototipo:</p> 	

Tabla 10. HU_4: Consultar control a clases

Fuente: Elaboración propia del autor

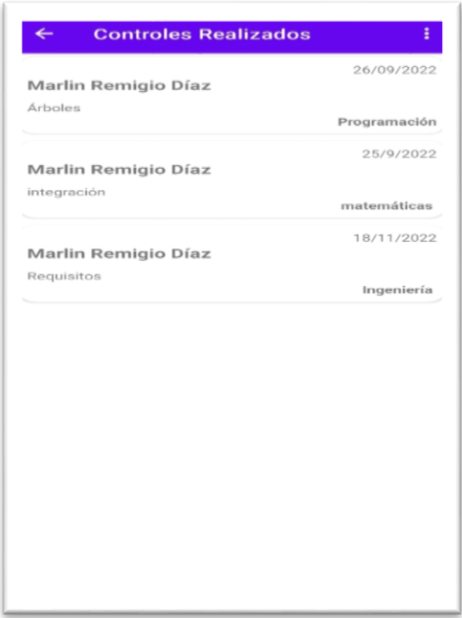
Historia de usuario	
Número: HU_4	Nombre: Consultar control a clases.
Prioridad en negocio: Media	
Descripción: Permite al usuario consultar un control a clases. El usuario va al listado de controles realizados y selecciona el control a clases que desea consultar y lo presiona, a continuación se muestra el contenido del control seleccionado que desea consultar.	
Prototipo:	 <p>El prototipo muestra una pantalla titulada 'Controles Realizados' con un menú de hamburguesa en la esquina superior derecha. La lista contiene tres ítems:</p> <ul style="list-style-type: none"> Marlin Remigio Díaz, 26/09/2022, Árboles, Programación Marlin Remigio Díaz, 25/9/2022, integración, matemáticas Marlin Remigio Díaz, 18/11/2022, Requisitos, Ingeniería

Tabla 11. HU_5: Buscar control a clases

Fuente: Elaboración propia del autor

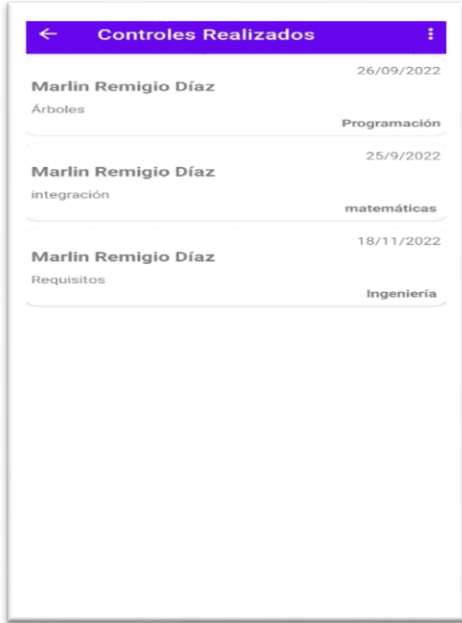
Historia de usuario	
Número: HU_5	Nombre: Buscar control a clases.
Prioridad en negocio: Media	
<p>Descripción: Permite al usuario hacer búsqueda de los controles a clases. El usuario va al listado de controles realizados y en la parte superior derecha de la interfaz se muestran tres puntos que al presionarlos se despliega un menú con diferentes filtros q serán útiles para efectuar la búsqueda más eficiente y ágil entre los filtros que se muestran están (Asignatura, Fecha, Tipo de clase y Categoría del profesor).</p>	
Prototipo:	 <p>The image shows a mobile application screen titled 'Controles Realizados'. It features a purple header with a back arrow and a menu icon. Below the header, there are three entries, each representing a control record. Each entry includes the name 'Marlin Remigio Díaz', a date, and a subject name. The first entry is dated 26/09/2022 with the subject 'Programación'. The second entry is dated 25/9/2022 with the subject 'matemáticas'. The third entry is dated 18/11/2022 with the subject 'Ingeniería'. Each entry also has a small category label on the left: 'Árboles', 'integración', and 'Requisitos' respectively.</p>

Tabla 12. HU_6: Listar controles a clases

Fuente: Elaboración propia del autor

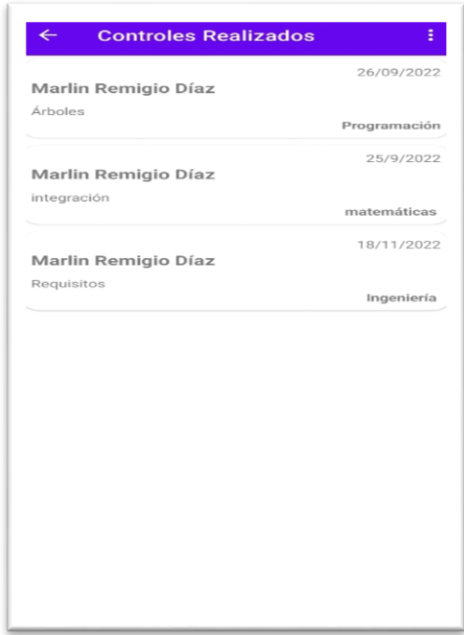
Historia de usuario	
Número: HU_6	Nombre: Listar controles a clases.
Prioridad en negocio: Media	
Descripción: Permite que en la medida en que se efectúen los controles a clases estos se vayan listando en orden descendente según la fecha de ejecución para tener así un orden lógico a la hora de filtrar.	
Prototipo:	 <p>El prototipo muestra una pantalla titulada 'Controles Realizados' con un menú de opciones. Las opciones listadas son:</p> <ul style="list-style-type: none"> Marlin Remigio Díaz (26/09/2022) - Árboles (Programación) Marlin Remigio Díaz (25/9/2022) - integración (matemáticas) Marlin Remigio Díaz (18/11/2022) - Requisitos (Ingeniería)

Tabla 13. HU_7: Realizar reportes de controles a clases

Fuente: Elaboración propia del autor

Historia de usuario	
Número: HU_7	Nombre: Realizar Reporte de Controles Realizados.
Prioridad en negocio: Media	
Descripción: Permite al usuario una vez que el control se encuentre en la zona de controles realizados hacer un reporte del mismo yendo a la parte superior derecha en los tres puntos q se muestran y presionado "Exportar Registros".	
Prototipo:	