

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
CONCEPCIÓN - CHILE



“DESARROLLO DE APLICACIÓN MÓVIL PARA
OPTIMIZAR LA GESTIÓN DEL CALENDARIO
ACADÉMICO UNIVERSITARIO”

DIEGO ANDRÉS LUCERO QUIERO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO EN INFORMÁTICA

Profesor Guía: Cristian Lara Valenzuela

Noviembre - 2024

DEDICATORIA

Dedicado a mi madre y a mi pareja.

El presente informe está escrito en honor a mi madre y mi pareja, ambas personas que me dieron las fuerzas, los ánimos, la paciencia y la comprensión suficiente para seguir adelante con mi carrera y mi vida en general, sin sus palabras de apoyo no habría sido posible haber completado esta importante etapa de mi vida. A ustedes, que siempre creyeron en mi a pesar de las adversidades, les dedico este trabajo con mi mas profundo agradecimiento.

A mis amigos, compañeros y seres queridos, con los que a lo largo de esta etapa compartimos momentos inolvidables, risas, momentos de estrés y alivio. Con su compañía, este corto viaje fue mas ameno y no tan estresante como podían describirlo ciertas personas.

A mis profesores, a aquellos quienes creyeron en nuestro potencial como estudiantes, su apoyo incondicional, su guía. La dedicación de ustedes ha dejado una marca grande en nuestros caminos.

A los que no se encuentran con nosotros en estos momentos, aquellos que a pesar de que no estén, siguen brindando su apoyo incondicional en cualquier parte del mundo, de la vida en general donde estén, gracias.

AGRADECIMIENTOS

Iniciar este agradecimiento es reconocer a la mujer extraordinaria que me dio la vida. Madre, tu amor incondicional ha sido mi faro constante, iluminando cada paso de este arduo camino hacia la obtención de mi título.

Tu presencia ha sido más que maternal; has sido mi consejera en los momentos más desafiantes de mi carrera, la voz que me alentó cuando la travesía parecía imposible. No solo fuiste madre, sino también padre, brindándome una fortaleza que va más allá de las palabras.

Gracias por tu dedicación y esfuerzo constantes. Eres la esencia de la mujer fuerte y entregada, y siempre serás mi gran amor. Este logro lleva tu marca indeleble, y cada página de este informe lleva consigo la huella de tu apoyo inquebrantable.

A mi increíble compañera de vida, mi pilar en momentos cruciales.

Quiero expresar mi profundo agradecimiento por los últimos cinco años que hemos compartido, llenos de altibajos que solo han fortalecido nuestra conexión. Tu apoyo inquebrantable ha sido mi fuente constante de ánimo y fortaleza, guiándome a través de los desafíos con una gracia que solo tú posees.

Eres más que mi soporte principal; eres mi inspiración constante. En los momentos en que las fuerzas flaqueaban, encontré en ti el impulso necesario para seguir adelante. Tu presencia ha iluminado los días oscuros y ha convertido los momentos difíciles en lecciones valiosas.

Eres la mujer que no solo comparte mi presente, sino que también me motiva a construir un futuro juntos. Gracias por ser mi roca, mi confidente y mi motivación constante. Este viaje ha sido más significativo y enriquecedor a tu lado, y estoy agradecido por cada instante.

A mis hermanas, familiares y amigos que me dieron el ánimo necesario.

Quiero agradecer a mis hermanas por ser quienes son, su hermandad a lo largo de los años me ha hecho darme cuenta que no todo es gris. Su cariño es mas que suficiente para saber que debo esforzarme mas por ser quien soy.

Quiero expresar mi agradecimiento de igual manera a mis familiares, en especial a mi tía, quien ha estado en los momentos que más necesito un abrazo, o una simple palabra de aliento. Gracias por ser parte de mi vida.

A mis profesores de enseñanza media, quienes hicieron darme cuenta que puedo más. Profesora Maritza Navarro y profesor Aldo Manquilef, quiero demostrar mi más sincero agradecimiento por demostrar y darme a entender que no soy una persona que se queda estancada cuando fracasa. Gracias por hacerme abrir los ojos y comprender que soy una persona que puede más.

RESUMEN

Resumen— Este proyecto aborda la integración del calendario académico de la Universidad Santa María mediante una aplicación móvil desarrollada en Kotlin con el patrón MVP. Se realizaron pruebas de usabilidad con mockups para validar la aceptación del usuario y la interfaz propuesta. Las tecnologías empleadas incluyen Android Studio, Retrofit y Picasso. Aunque la aplicación final está pendiente, la validación preliminar demuestra una mejora potencial en el acceso y visualización del calendario académico. La relevancia radica en ofrecer una solución práctica para optimizar la experiencia del usuario en la gestión de eventos académicos.

Palabras Clave— Palabras Clave: Integración de Calendario, Aplicación Móvil, Usabilidad, Kotlin, MVP, Retrofit, Picasso.

ABSTRACT

Abstract— This project addresses the integration of the academic calendar of Universidad Santa María through a mobile application developed in Kotlin using the MVP pattern. Usability testing with mockups was conducted to validate user acceptance and the proposed interface. Technologies employed include Android Studio, Retrofit, and Picasso. Although the final application is pending, preliminary validation demonstrates potential improvement in accessing and visualizing the academic calendar. The significance lies in providing a practical solution to enhance the user experience in managing academic events.

Keywords— Calendar Integration, Mobile Application, Usability, Kotlin, MVP, Retrofit, Picasso.

GLOSARIO

Aquí se deben colocar las siglas mencionadas en el trabajo y su explicación, por orden alfabético. Por ejemplo:

DI: Departamento de Informática.

UTFSM: Universidad Técnica Federico Santa María.

PK: Primary Key.

FK: Foreign Key.

USM: Universidad Santa María.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Universidad Federico Santa María.....	3
1.2 Calendario académico USM.....	3
1.3 Problemática actual.....	5
1.4 Rapidez y Accesibilidad en la Gestión del Calendario Académico.....	7
1.4.1 Tecnología y Organización diaria.....	7
1.5 Diversidad de Usuarios y Necesidades.....	8
1.6 Soluciones existentes.....	9
1.7 Consideraciones Tecnológicas.....	13
1.7.1 Desafíos tecnológicos.....	13
CAPÍTULO 2: MARCO CONCEPTUAL	15
2.1 Objetivos.....	16
2.2 Metodologías.....	16
2.3 Conceptos claves.....	19
2.4 Marco teórico.....	20
2.4.1 Kotlin.....	20
2.4.2 Android.....	23
2.4.3 Patrón MVP.....	23
2.4.4 Programación orientada a objetos.....	25
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	28
3.1 Descripción general de la solución propuesta.....	28
3.2 Casos de uso.....	29
3.3 Modelo de datos.....	44
3.3.1 Explicación de las tablas.....	44
3.4 Integración de Tecnologías.....	58
3.4.1 Introducción a las tecnologías.....	58
3.4.2 Detalles de Kotlin y Android Studio.....	59
3.4.3 Patrón de diseño MVP.....	60
3.4.4 Consideraciones de rendimiento y eficiencia.....	62

3.4.5	Manejo de errores y excepciones.....	63
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		65
4.1	Diseño de la interfaz de usuario.	65
4.1.1	Navegación en la aplicación.	65
4.2	Interacción con el calendario académico actual.	74
4.2.1	Acceso al calendario academico.....	74
4.2.2	Comparación entre aplicación y método tradicional.....	76
CAPÍTULO 5: CONCLUSIONES		78
5.1	Discusión:	79
5.2	Resultados y validación.	79
5.3	Contribución y aplicaciones del trabajo.	80
5.3.1	Identificación de Limitaciones Actuales:.....	80
5.3.2	Propuesta de Solución Innovadora:.....	80
5.3.3	Priorización del Modelo de Datos:	80
5.3.4	Modernización de Procesos Universitarios:.....	81
5.3.5	Guía Integral para Desarrolladores:.....	81
5.4	Impacto y recomendaciones.	81
ANEXOS		84
REFERENCIAS BIBLIOGRÁFICAS		85

ÍNDICE DE FIGURAS

1	Calendario académico USM.....	5
2	Calendario académico en su formato Excel.....	6
3	Google Calendar	9
4	Calendario de Outlook.	10
5	Google Keep.	11
6	Samsung Notes.	11
7	Recordatorios de iOS.....	12
8	Diagrama visualizar el calendario	29
9	Diagrama registrar actividad.....	30
10	Diagrama caso de uso notificar actividad.....	31
11	Diagrama caso de uso registrar perfil de usuario	33
12	Modelo de datos aplicación calendario académico UTFSM.	44
13	Relación de tabla persona.	45
14	Relación de la tabla actividad	45
15	Relación de la tabla act tipo	46
16	Relación de la tabla persona depto.....	47
17	Relación de la tabla externo	47
18	Relacion de la tabla ver actividades	48
19	Relacion de la tabla Departamento	49
20	Relacion de la tabla trab depto.....	50
21	Relacion de la tabla funcionario.....	51
22	Relacion de la tabla profesor.....	52
23	Relacion de la tabla Alumno.....	53
24	Relacion de la tabla Alumno carrera.....	54
25	Relacion de la tabla al carr cee.....	55
26	Relacion de la tabla cee.....	55
27	Relacion de la tabla carreras.....	56
28	Relacion de la tabla carreras.....	56
29	Relacion de las tablas depto academico, depto direccion y depto servicios.....	58
30	Inicio de sesión aplicación Calendario Académico USM.....	65
31	Registrar usuario aplicación Calendario Académico USM.	66
32	Calendario aplicación Calendario Académico USM.....	67
33	Calendario extendido aplicación Calendario Académico USM.	68
34	Agregar actividad a favoritos aplicación Calendario Académico USM.	69
35	Desplegar menú aplicación Calendario Académico USM.	70
36	Ingresar a favoritos aplicación Calendario Académico USM.....	71
37	Ingresar a perfil aplicación Calendario Académico USM.....	72
38	Ingresar a configuración aplicación Calendario Académico USM.	73
39	Búsqueda de eventos aplicación Calendario Académico USM.	74

ÍNDICE DE TABLAS

1	Caso de uso visualizar calendario académico.....	29
2	Caso de uso registrar actividad.....	30
3	Caso de uso notificar actividad.....	31
4	Caso de uso registrar perfil de usuario.....	32
5	Caso de uso registrar departamento.....	34
6	Caso de uso registrar usuario.....	35
7	Caso de uso inicio de sesión.....	36
8	Caso de uso añadir evento al calendario.....	37
9	Caso de uso filtrar evento.....	39
10	Caso de uso buscar evento.....	40
11	Caso de uso generar informe.....	41
12	Caso de uso compartir evento.....	42
13	Caso de uso agregar a favoritos.....	43

INTRODUCCIÓN

El calendario académico se erige como una herramienta fundamental en el proceso de estudios de los estudiantes universitarios, albergando información crucial, como las semanas de receso, fechas de inscripción de asignaturas, períodos para botar ramos, momentos para postular a becas universitarias, suspensiones de actividades docentes, y otros eventos relevantes. Esta valiosa fuente de información no solo impacta a los estudiantes, sino que también resulta esencial para los funcionarios en general de la Universidad.

Sin embargo, el gran desafío que enfrenta el calendario académico radica en su formato de entrega y la dificultad asociada para almacenarlo efectivamente en dispositivos electrónicos, ya sean celulares, computadores, notebooks o tablets. Actualmente, la Universidad Santa María ofrece la descarga del calendario en dos formatos principales: el formato principal, presentado como un archivo PDF, y el formato más reciente, presentado en Excel. Ambos formatos ofrecen la información necesaria para que la comunidad universitaria obtenga conocimiento sobre las actividades que marcan el transcurso del semestre.

Problema y contexto:

No obstante, en el contexto actual, el calendario académico enfrenta desafíos significativos relacionados con la accesibilidad y la usabilidad. La descarga de archivos PDF o Excel, aunque provee la información necesaria, presenta obstáculos en términos de practicidad y almacenamiento efectivo en los dispositivos de los usuarios.

Objetivo del proyecto:

El presente informe tiene como objetivo destacar la pertinencia de abordar esta problemática mediante la implementación de una solución específica: una aplicación móvil que integre de manera eficiente y accesible el calendario académico. A través de esta propuesta, se busca eliminar la carga actual asociada al manejo del calendario, ofreciendo una alternativa más dinámica y centrada en el usuario.

Justificación de la solución propuesta:

La adopción de una aplicación móvil dedicada al calendario académico se postula como una solución estratégica para mejorar la experiencia de los usuarios. Además de facilitar el acceso a la información, esta aplicación puede ofrecer funciones adicionales, como recordatorios personalizados, alertas de eventos importantes y una interfaz intuitiva que simplifique la navegación a través de las fechas académicas clave.

Conclusiones esperadas:

Se espera que este informe demuestre la viabilidad y la relevancia de implementar una aplicación móvil para integrar el calendario académico de la Universidad Santa María. Al abordar

la dificultad actual en el manejo de esta información, se busca ofrecer una solución que no solo alivie la carga de los usuarios, sino que también optimice la utilidad del calendario académico en el día a día universitario.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. Universidad Federico Santa María.

La Universidad Técnica Federico Santa María (USM) es una destacada institución de educación superior ubicada en Chile, con múltiples campus en lugares estratégicos como Valparaíso, Viña del Mar, Santiago y Concepción. Fundada en 1926, la USM ha construido una sólida reputación por ofrecer programas académicos de alta calidad, concentrándose especialmente en las áreas de ingeniería, ciencias tecnológicas y disciplinas afines.

La universidad se ha distinguido por su compromiso con la excelencia académica, la investigación de vanguardia y la formación de profesionales altamente capacitados. Su enfoque en la innovación y la transferencia de conocimiento la ha posicionado como una institución líder en el ámbito educativo y científico en Chile. Además, la USM fomenta la internacionalización, permitiendo a sus estudiantes participar en programas de intercambio y colaborar en proyectos de investigación a nivel global.

Con una infraestructura moderna y recursos especializados, la USM proporciona un entorno propicio para el aprendizaje y la investigación. Su compromiso con la responsabilidad social se refleja en iniciativas que buscan contribuir al desarrollo sostenible y al bienestar de la sociedad.

1.2. Calendario académico USM.

La UTFSM entrega una herramienta bastante importante como lo es el calendario académico, el cual tiene como destino organizar y visualizar fechas y eventos importantes, como lo pueden ser fechas para congelar, fechas para inscribir o des inscribir ramos, semanas de receso que se toma normalmente la Universidad para los estudiantes, entre otras fechas importantes y bastante relevantes dentro de la Universidad.

Esta herramienta es bastante relevante en base a la gestión eficiente del tiempo, tanto como para los estudiantes, profesores y el personal académico en general.

Comenzando con la optimización del aprendizaje, ya que para los estudiantes pueden ser de vital importancia las semanas de receso, teniendo así la capacidad de organizar sus estudios, asignar tiempos adecuados a cada materia y realizar tareas académicas de manera efectiva. Esto optimiza el proceso de aprendizaje, permitiendo a los estudiantes una comprensión mas profunda de los contenidos y mejorando el rendimiento académico.

La productividad de los docentes es un punto bastante importante, debido a que los pro-

Profesores y el personal académico se benefician de una gestión eficiente del tiempo al poder organizar y planificar sus clases, las investigaciones y actividades administrativas de manera más efectiva. Esto les permite centrarse en entregar una enseñanza de calidad, realizar investigaciones significativas y cumplir con sus responsabilidades administrativas sin sentirse abrumados.

La gestión eficiente del tiempo a nivel institucional contribuye a un mejor rendimiento general de la universidad. Permite la ejecución sin contratiempos de actividades académicas, administrativas y de investigación, fortaleciendo la reputación de la institución y atrayendo a estudiantes, profesores e investigadores de alta calidad.

Es importante mantener una buena gestión del estrés, por lo que una buena gestión del tiempo ayuda a reducir el estrés en todos los niveles. Estudiantes, profesores y personal académico pueden enfrentar sus responsabilidades de manera más equilibrada, evitando la acumulación de tareas y la presión excesiva.

La coordinación eficiente del tiempo facilita la colaboración entre estudiantes y profesores, así como entre diferentes departamentos y áreas de la institución. Esto es esencial para proyectos de investigación conjunta, eventos académicos y otras iniciativas que requieren la participación de diversos actores.

Disponer de tiempo de calidad permite a profesores y estudiantes dedicar esfuerzos a proyectos de investigación y desarrollo innovadores. La innovación es fundamental para el progreso académico y científico, así como para la relevancia continua de la institución en un entorno educativo en constante evolución.

1.3. Problemática actual.

Actualmente, la Universidad Técnica Federico Santa María (USM) ofrece dos formatos principales para acceder al calendario académico: el tradicional formato PDF y el nuevo formato en Excel. A pesar de la disponibilidad de estas herramientas, muchas personas no le dan la importancia necesaria, ya que, aunque el calendario en PDF busca ser "amigable" para el usuario, la experiencia de visualización no resulta del todo agradable. Este formato presenta el mes y año actual, seguido de los días, actividades o ceremonias, y finalmente, las actividades de docencia y los hitos reglamentarios (Ver figura 1).

ENERO 2023			
Día	Actividades o ceremonias	Docencia	Hitos reglamentarios
Lu. 02			Inicio periodo de inscripción de Asignaturas vía SIGA, para carreras de pregrado y programas de postgrado científico-tecnológico, correspondiente al 2023-1.
Vi. 06	Día del funcionario. Suspensión de actividades administrativas.		
Lu. 09	10:00 hrs. Ceremonia Titulación de Técnicos Universitarios e Ingenieros diurnos de Sede Concepción "Rey Balduino de Bélgica. Esta actividad se realiza en el Patio Cívico. 16:00 hrs. Ceremonia Titulación Ingenieros vespertinos de Sede Concepción "Rey Balduino de Bélgica. Esta actividad se realiza en el Patio Cívico.		

Figura 1: Calendario académico USM
Fuente: Vice-Rectoría Académica UTFSM

La universidad se esforzó por crear una nueva versión del calendario académico, si bien tiene todo lo que necesita para suplir la necesidad de tener que buscar fecha por fecha, este es más complicado de visualizar que el calendario académico actual, entregando un formato en Excel con los meses en las primeras casillas horizontales, los días del mes junto a su fecha correspondiente, y en la casilla de los días, un color que corresponde a las actividades que se llevaran a cabo en esta fecha (ver Figura 2).

Actualmente es el formato de calendario que menos se utiliza, ya que, se puede visualizar de peor manera que el calendario académico en su primera versión (Figura 1), y el hecho de tener que buscar en cada casilla distinta el color para ver que sucede en esa fecha del año, es una manera muy poco amigable de poder interactuar con el calendario académico.

DESARROLLO DE APLICACIÓN MÓVIL PARA OPTIMIZAR LA GESTIÓN DEL CALENDARIO ACADÉMICO
UNIVERSITARIO

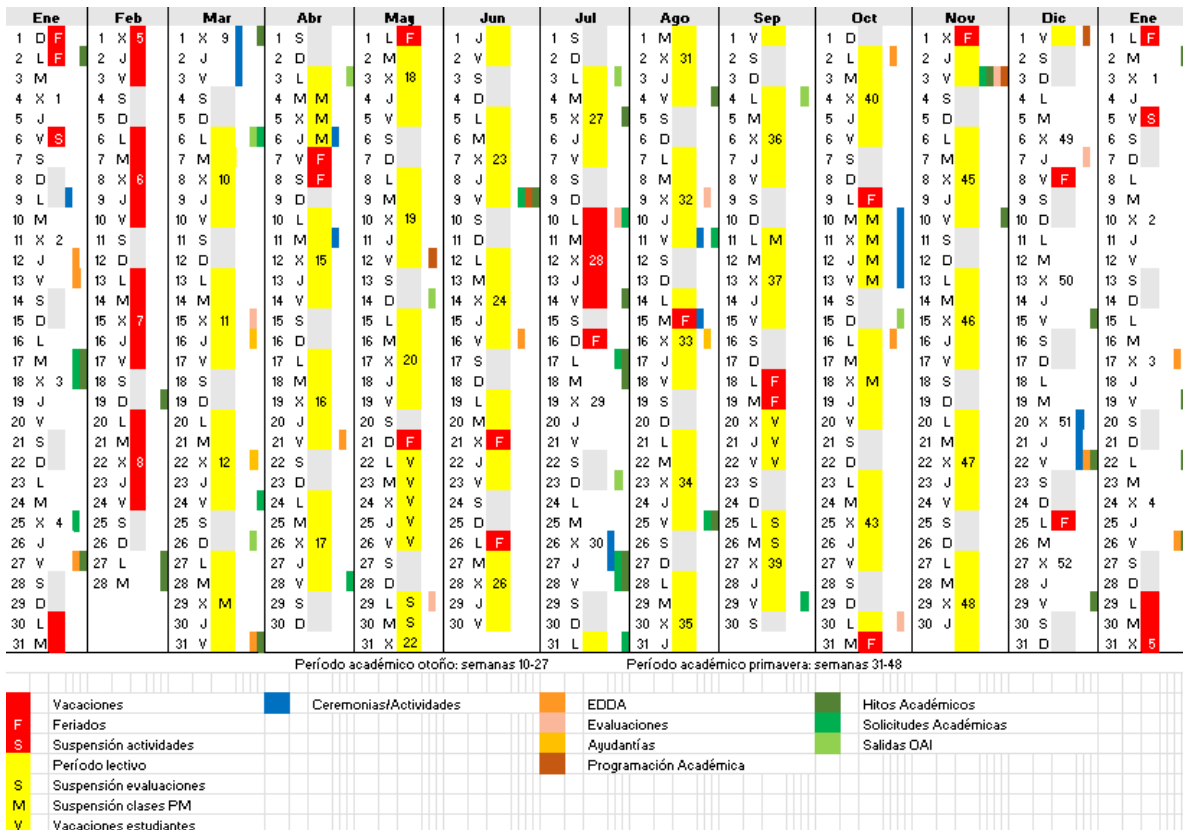


Figura 2: Calendario académico en su formato Excel

Fuente: Vice-Rectoría Académica UTFSM

Debido a la forma en que se presenta el calendario académico, los estudiantes pueden enfrentar dificultades para estar al tanto de fechas cruciales, como las semanas de receso. Estas pausas son esenciales para recargar energías y prepararse para las demandas académicas, como exámenes, entregas de trabajos e informes. Sin embargo, la falta de claridad en el calendario puede dificultar la planificación y la gestión del tiempo.

Además, muchos estudiantes pueden no estar al tanto de otras fechas importantes, como las fechas límite para congelar el año académico debido a problemas personales. La falta de información sobre estos hitos puede tener un impacto significativo en la planificación académica y el rendimiento general.

Otro aspecto crucial que puede pasar desapercibido es la falta de conocimiento sobre las fechas para postular a becas internas ofrecidas por la universidad. Dado que estas becas pueden ser una fuente importante de apoyo financiero, la gestión efectiva del calendario académico se vuelve aún más relevante.

En resumen, la necesidad de una gestión más efectiva del calendario académico es evidente, ya que impacta directamente en la capacidad de los estudiantes para enfrentar las demandas del semestre de manera organizada y eficiente.

1.4. Rapidez y Accesibilidad en la Gestión del Calendario Académico.

En la actualidad, los estudiantes universitarios enfrentan una carga académica significativa. Estudiar para exámenes, elaborar informes y preparar presentaciones para las diversas materias a lo largo de una carrera consume gran parte de su tiempo, manteniéndolos ocupados en estas actividades cruciales.

De manera similar, los funcionarios académicos, especialmente los profesores, enfrentan cargas laborales intensas. En muchos casos, los profesores no se limitan a impartir un solo curso, sino que pueden llegar a tener hasta 5 clases en un solo día. Esta situación limita considerablemente el tiempo que tienen para organizar revisiones de exámenes, evaluar trabajos o supervisar informes en general.

En este contexto, se ha identificado la necesidad de contar con un acceso fácil y rápido al calendario académico. Actualmente, buscar en internet el calendario académico de la universidad del año en curso resulta un proceso tedioso. La descarga y la posterior pérdida en archivos son inconvenientes comunes, lo que provoca una pérdida de tiempo valioso tanto para estudiantes como para profesores. Recuperar este tiempo a menudo resulta desafiante, afectando las actividades cotidianas de ambos grupos.

1.4.1. Tecnología y Organización diaria.

En la actualidad, el mundo ha experimentado una amplia adaptación y modernización en diversos aspectos. La mayoría de las personas posee un dispositivo móvil con el cual interactúa diariamente, independientemente de la hora. Esta preferencia por el uso de dispositivos móviles se debe principalmente a su comodidad. Los teléfonos celulares son más prácticos de llevar consigo en comparación con dispositivos más grandes, como computadoras portátiles o notebooks. Su tamaño compacto facilita el transporte y manejo, convirtiéndolos en una elección conveniente para la mayoría de las personas.

La tecnología ha tomado un papel importante como simple facilitadora y se ha convertido en un elemento central para la organización de nuestras vidas. Desde la comunicación hasta la gestión del tiempo, la dependencia de dispositivos electrónicos y plataformas digitales se ha vuelto omnipresente. La interconexión de dispositivos, la disponibilidad instantánea de información y la comunicación sin fronteras han redefinido la manera en que llevamos a cabo nuestras rutinas diarias.

En el ámbito laboral, la tecnología ha permitido una flexibilidad sin precedentes. Las herramientas de colaboración en línea, las videoconferencias y las aplicaciones de gestión de tareas han alterado la forma en que trabajamos, permitiéndonos organizar nuestras responsabilidades de manera más eficiente y colaborar con colegas en cualquier parte del mundo. Sin embargo, esta conectividad constante también ha borrado las líneas entre la vida profesional y personal, desafiando la noción tradicional de equilibrio entre trabajo y vida.

La dependencia de la tecnología también se manifiesta en la forma en que gestionamos nuestras vidas personales. Aplicaciones de organización, recordatorios automáticos, asistentes virtuales y dispositivos inteligentes para el hogar han simplificado tareas que alguna vez requerían un esfuerzo consciente. Si bien estas innovaciones han mejorado la eficiencia, también plantean preguntas sobre la dependencia excesiva y la pérdida de habilidades básicas de organización sin la ayuda de la tecnología.

Con el objetivo de ofrecer una solución fácil y rápida, se ha optado por desarrollar la aplicación "Calendario Académico UTFSM". Esta aplicación está diseñada para que tanto estudiantes como personal académico puedan visualizar de manera amigable y accesible el calendario académico correspondiente al año en curso de la Universidad. La aplicación busca simplificar la vida de los usuarios que hagan uso de esta herramienta a menudo al proporcionar una herramienta fácil de usar con funciones que les permitan recordar fechas importantes de manera rápida y eficiente.

Con "Calendario Académico UTFSM", los estudiantes y personal académico ya no necesitarán llevar consigo el calendario en formato PDF o Excel en sus dispositivos. La aplicación ofrece una forma más sencilla de acceder a la información del calendario, evitando la molestia de buscar fechas específicas en documentos extensos. Con esta aplicación, se busca aliviar a los estudiantes de la carga de tener que buscar durante mucho tiempo una fecha en un documento PDF o en el formato actual del calendario académico, facilitando así su experiencia académica.

1.5. Diversidad de Usuarios y Necesidades.

Esta propuesta de aplicación del calendario académico no está solamente limitada a satisfacer las necesidades de los estudiantes, sino que también está diseñada para ofrecer una solución valiosa para un amplio espectro de usuarios dentro de la comunidad universitaria. Los beneficios abarcan a funcionarios, profesores y personal administrativo, quienes a menudo se enfrentan al mismo problema de olvidar fechas críticas y eventos importantes.

Los profesores por ejemplo, desempeñan un papel fundamental en la universidad y están sometidos a plazos estrictos, como la fecha límite para subir las calificaciones en la plataforma del SIGA. La aplicación del calendario académico también les brindará una herramienta efectiva para recordar estos plazos y evitar retrasos innecesarios. Los profesores tendrán la capacidad de configurar recordatorios personalizados, permitiéndoles recibir avisos con anticipación sobre fechas cruciales, como la fecha límite para cargar las notas. Pueden elegir cuántos días antes desean ser recordados, lo que optimiza su gestión del tiempo y reduce riesgos de olvido.

Además de los profesores, otros funcionarios administrativos, como secretarios y personal de finanzas, también serán beneficiados enormemente por la aplicación. Podrán utilizarla para hacer seguimiento de sus propias tareas y plazos importantes, como la presentación de

informes financieros trimestrales o la organización de eventos institucionales. La aplicación proporcionara una herramienta centralizada y fácil de usar para gestionar eventos y fechas críticas en el entorno académico.

La aplicación del calendario académico tiene como objetivo entregar una solución integral que mejora la gestión del tiempo y la organización de todos los miembros de la comunidad universitaria. Su versatilidad y capacidad de adaptación a las necesidades individuales de cada usuario la convierte en una herramienta esencial para mantenerse al tanto de las fechas y eventos importantes en la vida universitaria.

1.6. Soluciones existentes.

Existen soluciones actualmente para generar recordatorios, como lo pueden ser google calendar, el calendario de Outlook, Google Keep, Samsung Notes o Samsung Reminder (Para dispositivos Android), Reminders (Dispositivos iOS).

Google Calendar: Es una aplicación de calendario en línea, permite organizar eventos, recordatorios y tareas, sus funciones en general son la creación de eventos y compromisos, vistas de calendario (Día, semanas, mes y agenda), notificaciones, compartir calendarios, colores y etiquetas, búsqueda de eventos, sincronización con multiplataforma, acceso desde cualquier lugar, recordatorios con asistente de google.

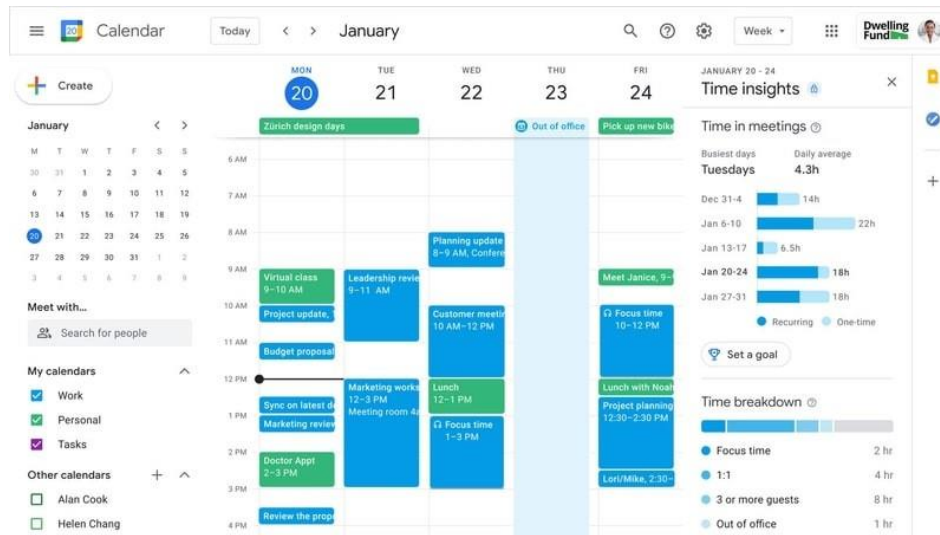


Figura 3: Google Calendar.
Fuente: Orlando Osorio.

Calendario de Outlook: Al igual que Google Calendar, es una aplicación de calendario en línea que se puede descargar desarrollada por Microsoft, tiene funciones como la creación de eventos, vistas del calendario (día, semana, mes y agenda), invitaciones y reuniones, recordatorios y alertas, integración con correo electrónico, categorización y colores, compartir calendarios, tareas y recordatorios, sincronización con dispositivos móviles, búsqueda de eventos, acceso en la nube.

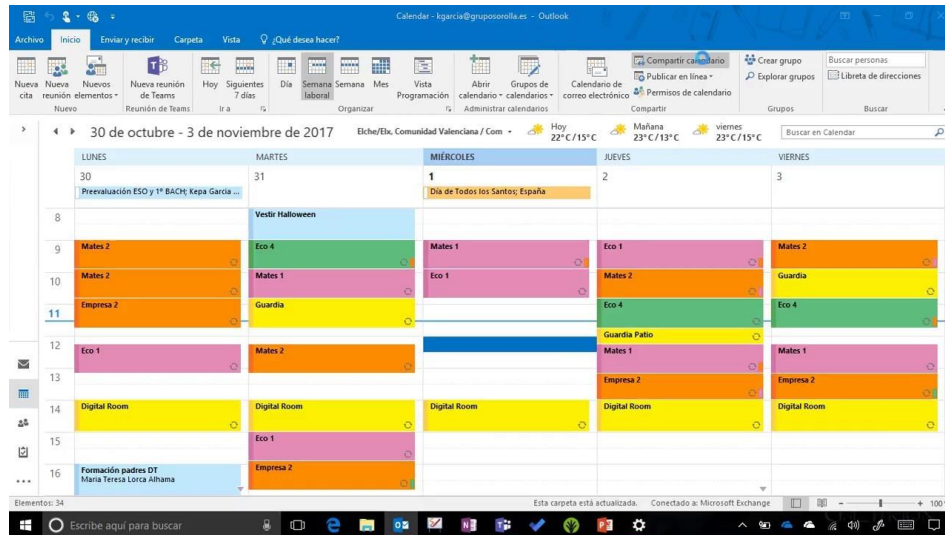


Figura 4: Calendario de Outlook.
Fuente: Canal de Youtube profeconomia.

Google Keep: Es una aplicación de notas y listas desarrollada por google, tiene funcionalidad como la creación de notas, listas de tareas, color y etiquetas, recordatorios, captura de imágenes y notas de voz, búsqueda y filtros, sincronización en la nube, integración con google drive, compartir notas, acceso desde dispositivo móvil y web, eliminar notas o archivarlas.

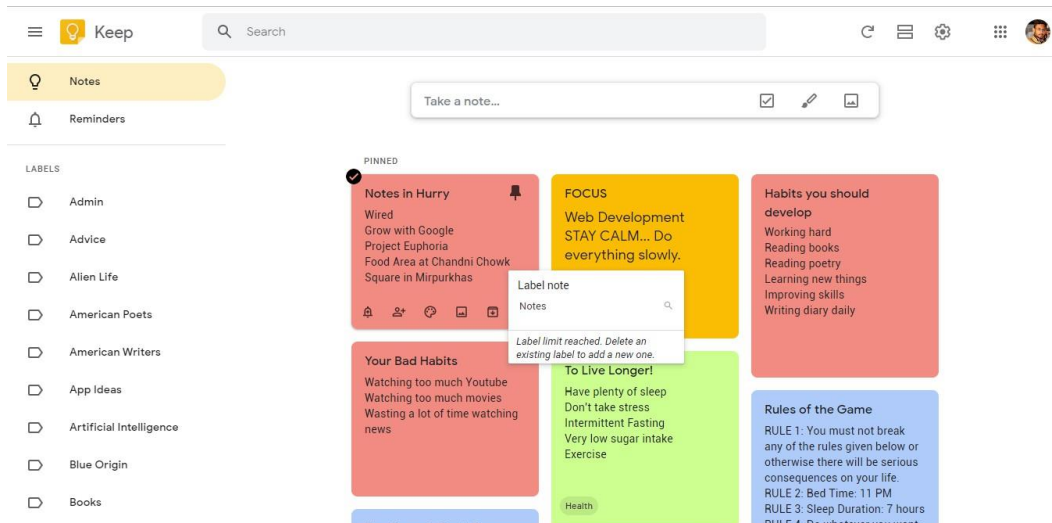


Figura 5: Google Keep.
Fuente: Software Advice.

Samsung notes: Es una aplicación creada por Samsung diseñada para tomar notas y realizar otras funciones relacionadas con la creación de contenido, tiene ciertas funcionalidades como la creación de notas, agregar dibujos, imágenes, grabación de voz y pegatinas a las notas, sincronización con la nube, búsqueda de notas, organización y carpetas, edición de notas, dibujar y escribir a mano, exportación e impresión, seguridad y bloqueo de notas, recordatorios y lista de tareas.

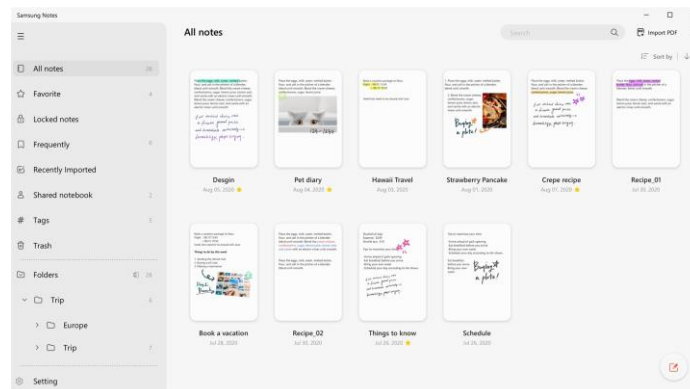


Figura 6: Samsung Notes.
Fuente: Imagen de Microsoft App Store.

Recordatorios (iOS): Es una aplicación móvil desarrollada por Apple para ayudar a realizar seguimientos de tareas, recordatorios y listas de cosas por hacer, tiene funciones como la creación de recordatorios, listas de tareas, fecha y hora de vencimiento de las tareas, recordatorios en base a la ubicación, integración con Siri, notificar recordatorio, colaboración con otros usuarios, integración con otras aplicación de Apple como calendario y mensajes, organización y edición, sincronización con iCloud.

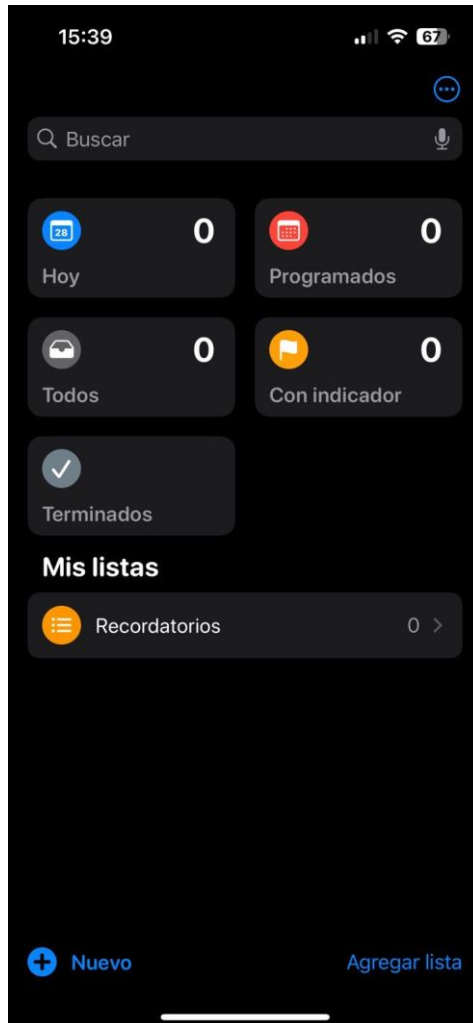


Figura 7: Recordatorios de iOS.

Fuente: Imagen tomada desde dispositivo Apple.

Si bien las aplicaciones mencionadas son útiles en el día a día, ninguna se compara directamente con una diseñada específicamente para el calendario académico de la USM. A pesar de contar con funciones como guardar actividades importantes, recordar eventos marcados como importantes, eliminar recordatorios, entre otras, muchos usuarios pueden no tomarse el tiempo de ingresar manualmente cada actividad. Este proceso podría consumir más tiempo del que ahorraría en comparación con buscar el calendario académico en sus formatos actuales en internet y simplemente guardarlo.

La solución propuesta busca integrar todas estas funciones ya presentes en otras aplicaciones, como agregar y eliminar recordatorios o buscar eventos. La distinción clave es que no se requerirá que los usuarios ingresen manualmente cada recordatorio, ya que la aplicación del calendario contendrá previamente cargado el calendario académico del año actual. Esto no solo ahorra tiempo a los usuarios que deseen agregar recordatorios sobre fechas importantes, sino que también permite que la aplicación notifique automáticamente sobre fechas importantes para todos los usuarios, eliminando la necesidad de que cada usuario marque esas fechas como importantes, ya que serían relevantes para todos.

1.7. Consideraciones Tecnológicas.

La implementación de esta solución requerirá el dominio de lenguajes de programación específicos como Kotlin para Android o Swift para iOS. Además, será esencial comprender y aplicar patrones de diseño, especialmente el Modelo-Vista-Presentador (MVP). Un conocimiento sólido de programación orientada a objetos también será fundamental para desarrollar de manera eficiente la lógica y la estructura de la aplicación.

Adicionalmente, se necesitarán habilidades en el manejo de bases de datos para crear y gestionar la base de datos que almacenará todos los datos relevantes para la aplicación del calendario académico. Este aspecto incluye la capacidad para diseñar una estructura de base de datos eficiente y segura, así como implementar operaciones de lectura y escritura de datos de manera óptima.

1.7.1. Desafíos tecnológicos.

Desarrollar aplicaciones iOS con Swift en Windows presenta desafíos debido a varias razones fundamentales:

- Xcode y macOS: Xcode, el entorno de desarrollo integrado (IDE) de Apple, es esencial para el desarrollo de aplicaciones iOS con Swift. Sin embargo, Xcode está disponible exclusivamente para macOS. Por lo tanto, si bien puedes escribir código Swift en otros entornos, necesitarás Xcode para compilar y depurar aplicaciones iOS, y esto limita el desarrollo a un entorno Mac.

- **Requisitos del sistema operativo:** La creación de aplicaciones iOS con Swift requiere el uso de funciones específicas de macOS y depende de las últimas versiones de Xcode, que están diseñadas para ejecutarse en sistemas operativos macOS.
- **Integración con herramientas de Apple:** Xcode y otras herramientas proporcionadas por Apple están estrechamente integradas en el ecosistema macOS. Estas herramientas trabajan en conjunto para facilitar el desarrollo, la depuración y las pruebas de aplicaciones iOS. Intentar replicar este entorno en Windows sería una tarea compleja debido a las dependencias y la integración específica del sistema operativo.
- **Apple tiene un control estricto sobre el hardware y el software de sus dispositivos.** Esto incluye la optimización de Swift para el ecosistema Apple y la garantía de que las aplicaciones desarrolladas en este entorno funcionen de manera eficiente en los dispositivos iOS.

Aunque no es posible desarrollar directamente aplicaciones iOS en Windows, hay algunas alternativas y enfoques que los desarrolladores pueden considerar:

- **Utilizar un Mac o un servicio en la nube:** Es posible utilizar un Mac físico para el desarrollo o explorar servicios en la nube que ofrecen acceso a un entorno macOS para desarrollo.
- **Plataformas de Desarrollo Cruzado:** Algunas herramientas permiten el desarrollo de aplicaciones utilizando un código base que se puede desplegar en diferentes plataformas, incluidas iOS y Android. Ejemplos de estas herramientas incluyen Flutter (Dart) y Xamarin (C Sharp).

Aunque estas alternativas pueden ser útiles en ciertos casos, para el desarrollo de aplicaciones iOS de alta calidad y plenamente compatible, se recomienda utilizar un entorno macOS con Xcode y Swift.

CAPÍTULO 2

MARCO CONCEPTUAL

El calendario académico desempeña un papel fundamental en la vida de cualquier institución educativa, ya que este es la hoja de ruta que guía a los estudiantes, a los profesores y a todo el personal administrativo a lo largo del año académico, sin embargo, a pesar de la importancia que tiene el calendario académico, presenta numerosos desafíos que limitan su utilidad y eficacia debido a la entrega de este que actualmente se encuentra en formato de PDF y en Excel.

Actualmente el calendario en formato PDF y Excel es notoriamente poco estético y nada interactivo. Una vez los descarga el usuario son olvidados en su mayoría o pasan desapercibidos en cualquiera de los dispositivos que tenga el usuario. No permiten la actualización de información, lo que conlleva a la obsolescencia de fechas y eventos. Esto significa que, si hay cambios en las fechas de eventos o actividades académicas, como modificaciones en el horario de las clases, cancelación de conferencias o cambios en las fechas de los exámenes, la versión del calendario que se encuentra en el dispositivo del usuario seguirá mostrando la información desactualizada. Esto causa principalmente confusiones en los usuarios, ya que pueden disponer del calendario desactualizado y perderse eventos importantes o en algunos casos recibir información incorrecta sobre fechas y actividades académicas importantes. Además, carecen de la capacidad de generar recordatorios personalizado para ciertas fechas que el usuario estime importantes, lo que puede llevar a pérdida de plazos críticos.

La gestión de la información académica también puede resultar desafiante. Los departamentos académicos y el centro de estudiante muchas veces luchan por difundir información relevante de manera oportuna y eficaz, lo que puede dar a lugar a una comunicación fragmentada y a la dispersión de datos importantes en múltiples canales, como grupos de WhatsApp, redes sociales o correos electrónicos.

Ante estos desafíos, se plantea la necesidad de una solución que no solo modernice la forma en que se presenta el calendario académico, sino que también mejore la interacción, la comunicación y la utilidad de esta herramienta vital. Este proyecto tiene como objetivo abordar estas limitaciones y brindar una solución integral mediante el desarrollo de una aplicación móvil diseñada para transformar la gestión del calendario académico.

2.1. Objetivos

El objetivo central de este proyecto es reemplazar el calendario académico en sus formatos actuales por una aplicación móvil que ofrezca una experiencia más eficiente y enriquecedora.

El objetivo general es: Desarrollar una aplicación móvil que integre los distintos calendarios académicos, de beneficio y otros de interés, con los que cuenta la Universidad para proporcionarle a los estudiantes, docentes y al personal administrativo una herramienta que sea eficiente y a la vez accesible para la gestión y el seguimiento de las actividades y eventos académicos de la universidad.

Los objetivos específicos incluyen:

1. Analizar el formato del calendario académico, beneficios estudiantiles y de otro interés para una mejor gestión de la vida universitaria.
2. Definir los procesos de funcionalidad de la aplicación
3. Diseñar la aplicación, considerando los calendarios académicos estudiados y la funcionalidad definida.
4. Implementar la aplicación según las especificaciones de diseño.
5. Diseñar una interfaz de usuario altamente intuitiva y atractiva que sea fácil de utilizar y visualmente agradable.
6. Desarrollar funcionalidades que permitan a los usuarios agregar, editar y eliminar eventos en el calendario, así como configurar recordatorios personalizados.
7. Integrar la aplicación con los departamentos académicos y el centro de estudiantes para una comunicación eficiente y oportuna de información relevante.
8. Facilitar la personalización del calendario por parte de los usuarios, permitiéndoles adaptar la aplicación a sus necesidades específicas.

2.2. Metodologías

Para el primer objetivo necesitamos:

- Realizar una revisión exhaustiva de los calendarios académicos actualmente existentes en sus distintos formatos, analizando su estructura y contenido.
- Realizar encuestas y entrevistas con estudiantes, profesores y personal académico en general para recopilar opiniones y experiencias sobre el uso actual de los calendarios.

- Identificar las deficiencias clave del formato actual y documentar los beneficios potenciales de una aplicación móvil en términos de accesibilidad, actualización de datos y facilidad de uso.

Para el segundo objetivo necesitamos:

- Realizar talleres con las partes interesadas, con representantes de departamentos académicos y el centro de estudiantes para identificar sus necesidades específicas y procesos de gestión de eventos.
- Documentar los procesos y definir los flujos de trabajo y procesos de gestión de eventos que serán fundamentales en la aplicación.
- Documentar los requisitos funcionales y no funcionales necesarios para el desarrollo de la aplicación en base a los procesos identificados.

Para el tercer objetivo necesitamos:

- Crear un diseño de información que incluya la estructura de la aplicación y las relaciones entre las diferentes secciones.
- Desarrollar prototipos de alta fidelidad de la interfaz de usuario para visualizar como se vera y funcionara la aplicación.
- Realizar pruebas de usabilidad con usuarios reales para evaluar la efectividad del diseño y obtener retroalimentación.
- Ajustar el diseño de la aplicación en función de los comentarios de los usuarios y los resultados de la prueba de usabilidad.

Para el cuarto objetivo necesitamos:

- Elegir las tecnologías adecuadas para el desarrollo de la aplicación, teniendo en cuenta las plataformas móviles objetivo (Android, iOS).
- Implementar la aplicación de manera iterativa, enfocándose en las funcionalidades esenciales primero.
- Realizar pruebas regulares durante el desarrollo para identificar y solucionar problemas técnicos y funcionales.

Para el quinto objetivo necesitamos:

- Utilizar herramientas de diseño gráfico para crear una interfaz visualmente atractiva y coherente.
- Realizar pruebas de usabilidad a lo largo del proceso de diseño para evaluar la experiencia del usuario y realizar ajustes según sean necesarios.
- Aplicar principios de diseño centrado en el usuario para garantizar que la interfaz sea fácil de usar y agradable visualmente.

Para el sexto objetivo necesitamos:

- Definir y diseñar las funciones relacionadas con la gestión de eventos y recordatorios.
- Implementar estas funcionalidades y realizar pruebas exhaustivas para garantizar que funcionen de manera confiable y que cumplan con los requisitos.

Para el séptimo objetivo necesitamos:

- Colaborar con los departamentos y el centro de estudiantes para establecer protocolos de comunicación y definir los datos que deben integrarse.
- Crear API's que permitan la transferencia de información entre la aplicación y los sistemas internos de la universidad.
- Realizar pruebas exhaustivas para garantizar una comunicación eficiente y oportuna de información relevante.

Para el octavo y último objetivo necesitamos:

- Diseñar una sección de configuración en la aplicación permita a los usuarios personalizar su calendario.
- Desarrollar estas opciones de personalización y realizar pruebas de usuario para asegurarse de que sean efectivas y fáciles de usar.

2.3. Conceptos claves

1. **Calendario Académico:** Es un registro estructurado de eventos y fechas importantes relacionadas con la vida académica de una universidad, como el inicio y fin de clases, exámenes, fechas límites de inscripción, vacaciones, entre otros eventos relevantes.
2. **Usabilidad:** Se refiere a la facilidad con la que los usuarios pueden interactuar con la aplicación móvil. Una aplicación usable es aquella que es intuitiva y eficiente de usar, lo que mejora la experiencia del usuario.
3. **Actualización en tiempo real:** Es la capacidad que tiene la aplicación para reflejar información del calendario académico de manera instantánea y precisa, de modo que los usuarios siempre tengan acceso a datos actualizados.
4. **Comunicación efectiva:** La capacidad de la aplicación para facilitar la comunicación eficiente entre los usuarios y los departamentos académicos, asegurando que la información crítica se comparta de manera oportuna.
5. **Desarrollo en Kotlin** Kotlin es un lenguaje de programación moderno y versátil que se utiliza ampliamente para el desarrollo de aplicaciones móviles Android. Se caracteriza por su concisión y seguridad de tipo, lo que lo convierte en una elección popular entre los desarrolladores Android.
6. **Android Studio** Es el IDE oficial para el desarrollo de aplicaciones Android. Proporciona herramientas y recursos esenciales para escribir, depurar y compilar aplicaciones en Kotlin.
7. **Programación orientada a objetos:** Kotlin es un lenguaje de programación orientada a objetos, lo que significa que se basa en conceptos como clases, objetos y herencia para organizar y estructurar el código de la aplicación.
8. **Patrón MVP:** Es una arquitectura de diseño de software que separa claramente las responsabilidades dentro de una aplicación móvil. Comprende tres componentes principales: El Modelo (Model), la Vista (View) y el Presentador (Presenter).
9. **Model:** El Modelo representa los datos y la lógica de negocio de la aplicación. El modelo gestiona la obtención y almacenamiento de eventos académicos, fechas importantes y otra información relacionada con el calendario.
10. **View:** La Vista corresponde a la interfaz del usuario de la aplicación. Es responsable de mostrar la información al usuario y de recibir las interacciones del usuario. Se encarga de mostrar el calendario, eventos y permitir a los usuarios interactuar con la aplicación.
11. **Presenter:** El Presentador actúa como intermediario entre el Modelo y la Vista. Es el responsable de gestionar la lógica de presentación y la interacción entre estos dos componentes. Maneja la actualización de la Vista con los datos del Modelo y responde a las acciones del usuario.

2.4. Marco teórico

El desarrollo de aplicaciones móviles ha experimentado un crecimiento significativo en los últimos años, convirtiéndose en una herramienta esencial para facilitar la gestión y organización de diversas actividades. En el contexto académico, la gestión eficiente del calendario académico es esencial para estudiantes, profesores y el personal administrativo de la universidad. Como se ha dicho durante el desarrollo del presente documento, el calendario académico en su formato actual tiene un gran problema, los usuarios de la Universidad no le toman el peso significativo que este tiene, por lo que se ha optado a crear una aplicación móvil la cual reemplazara el uso del calendario académico en formato PDF y en su formato de Excel. Para esto necesitamos conocer las herramientas que se utilizaran para el desarrollo de esta aplicación.

2.4.1. Kotlin

Es un lenguaje de programación dirigido a la plataforma Java. Kotlin es conciso, seguro, pragmático y se enfoca en la interoperabilidad con código Java. Puede ser utilizado prácticamente en cualquier lugar donde se utiliza Java hoy en día: para el desarrollo en el lado del servidor, aplicaciones Android y mucho más. Kotlin funciona muy bien con todas las bibliotecas y marcos de Java existentes y se ejecuta con el mismo nivel de rendimiento que Java. [Jemerov e Isakova, 2017]

Kotlin es un lenguaje de programación de uso general, que encuentra su principal aplicación en el desarrollo de aplicaciones móviles para la plataforma de Android. Sin embargo, las capacidades de Kotlin no se limitan únicamente a las aplicaciones para Android, sino que también extiende su uso a otra áreas incluyendo:

- Desarrollo en el lado del servidor: Tradicionalmente, el desarrollo de aplicaciones web en el lado del servidor ha empleado el lenguaje Java. Kotlin puede utilizarse en conjunto con Java para desarrollar aplicaciones en el servidor, ya que es compatible con las bibliotecas de Java.
- Desarrollo web full-stack: Los desarrolladores aprovechan Kotlin para JavaScript con el fin de traducir líneas de código de Kotlin a JavaScript en el desarrollo web en el lado del cliente. Esta aproximación les permite usar el mismo código tanto en el lado del cliente como en el servidor.
- Desarrollo móvil multi plataforma: Kotlin se emplea en el desarrollo de aplicaciones móviles no solo para Android, sino también para otras plataformas como iOS, watchOS y Linux.

- Ciencia de datos: Kotlin se utiliza en tareas de ciencia de datos, como la creación de flujos de datos y la implementación de modelos de aprendizaje automático en entornos de producción. Además, herramientas populares de ciencia de datos como Jupyter y Zeppelin, que se utilizan con Java, también ofrecen soporte para Kotlin.

Kotlin tiene algunos beneficios al utilizarse los cuales son los siguientes:

1. Interoperabilidad: En el contexto de la programación, la interoperabilidad se refiere a la capacidad de distintos lenguajes o tecnologías para trabajar juntos de manera efectiva. En el caso de Kotlin, es digno de destacar que presenta un alto nivel de interoperabilidad con Java. Esto se debe a que ambos lenguajes pueden compilar su código fuente en el mismo código de bytes, lo que significa que pueden coexistir y utilizarse de manera conjunta en un mismo proyecto. Además de su interoperabilidad con Java, ofrece la posibilidad de compilar su código en otros contextos. Por ejemplo, es capaz de generar código JavaScript, lo que facilita su uso en aplicaciones web y desarrollo front-end. También es compatible con un compilador LLVM, lo que habilita a los programadores a realizar compilaciones en tiempo real, asegurando que el código integrado en otros programas se ejecute sin contratiempos. Una ventaja adicional es que Kotlin comparte herramientas y recursos con Java, o sea que los desarrolladores familiarizados con Java pueden adaptarse con facilidad al lenguaje de Kotlin, lo que simplifica la migración de aplicaciones previamente desarrolladas en Java a Kotlin. En resumen, estas características de Kotlin hacen que la migración de aplicaciones desde Java hacia Kotlin sea un proceso más fluido y eficiente. Los programadores pueden aprovechar las ventajas de Kotlin, como su sintaxis más concisa y su enfoque en la seguridad del tipo, mientras continúan utilizando código existente en Java. Esto resulta particularmente útil en el entorno académico y empresarial, donde la adaptabilidad y la eficiencia en el desarrollo de software son aspectos cruciales.
2. Seguridad: El lenguaje de programación Kotlin ha sido meticulosamente diseñado con el propósito de mitigar errores de codificación habituales que pueden comprometer la integridad del código o dejar potenciales vulnerabilidades en el sistema. Uno de los aspectos más destacados de Kotlin en este contexto es su enfoque en la seguridad en relación con los valores nulos, lo que se traduce en la eliminación de errores de excepción de puntero nulo.

En otras palabras, Kotlin ha sido concebido para brindar una capa adicional de robustez y fiabilidad al proceso de desarrollo de software. Esta característica es particularmente relevante en el mundo de la programación, donde la gestión inadecuada de valores nulos es una fuente común de errores críticos. La incorporación de Kotlin en proyectos de desarrollo no solo puede evitar desafíos técnicos innecesarios, sino que también contribuye a la seguridad y la estabilidad del software, un aspecto de importancia fundamental tanto en la academia como en la industria.

3. **Claridad:** Kotlin se destaca por su capacidad de reducir la redundancia inherente en la sintaxis básica de lenguajes ampliamente utilizados como Java. Este enfoque se traduce en un ahorro de tiempo significativo para los desarrolladores, ya que Kotlin permite generar un código más conciso y eficiente. En consecuencia, los programadores tienen la posibilidad de crear programas con una cantidad considerablemente menos de código redundante, lo que, en última instancia, se traduce en un aumento notable de su productividad.

Este aspecto de claridad y eficiencia en la escritura de código es de gran relevancia tanto en el ámbito académico como en la industria, ya que no solo simplifica la tarea del desarrollo, sino que también contribuye a la mantenibilidad y legibilidad del código, factores críticos en el ciclo de vida del software.

4. **Soporte de herramientas:** Es importante destacar que Kotlin cuenta con un sólido respaldo en términos de herramientas provenientes del ecosistema de Android. Estas herramientas están diseñadas y optimizadas específicamente para el desarrollo en plataforma Android. Entre ellas, podemos mencionar a Android Studio, Android KTX y Android SDK.

Este respaldo de herramientas es de suma importancia para los desarrolladores, ya que les permite trabajar de manera más eficiente y efectiva en el desarrollo de aplicaciones Android utilizando Kotlin como lenguaje de programación. Estas herramientas no solo simplifican el proceso de codificación, sino que también brindan capacidades avanzadas de depuración, análisis y optimización, lo que contribuye significativamente a la calidad y el rendimiento de las aplicaciones móviles. En el contexto académico y profesional, la disponibilidad de herramientas de alto nivel como estas juega un papel importante en el éxito y la eficacia de los proyectos de desarrollo software.

5. **Apoyo de la Comunidad:** A pesar de ser un lenguaje de programación relativamente joven en comparación con Java, Kotlin ha logrado establecer una comunidad de desarrolladores sumamente activa. Esta comunidad se dedica incansablemente a perfeccionar el lenguaje y aportar recursos de documentación valiosos para su adopción y desarrollo continuo.

Esta colaboración y compromiso de la comunidad son elementos fundamentales en el mundo de la programación. En el caso de Kotlin, los miembros de la comunidad trabajan en diversas áreas, incluyendo la identificación y corrección de errores, la creación de bibliotecas adicionales y la producción de documentación detallada.

Para los estudiantes y profesionales que se aventuran en el aprendizaje y uso de Kotlin, el apoyo de la comunidad es un activo invaluable. Proporciona un respaldo esencial para abordar desafíos técnicos, promover mejores prácticas y mantenerse actualizado en un campo tan dinámico como la programación. Además, esta comunidad colaborativa también refleja el espíritu de innovación y desarrollo continuo que caracteriza a la industria de la tecnología, lo que hace que Kotlin sea una elección sólida tanto en el ámbito académico como en el profesional.

[Lutkevich, 2022]

2.4.2. Android

Android, como sistema operativo móvil, se basa en una versión personalizada del kernel de Linux y tiene sus raíces en la adquisición de la empresa Android, Inc. por parte de Google en 2005. Desde entonces, Google ha liderado el desarrollo de Android y lo ha mantenido como un sistema operativo de código abierto y gratuito, bajo la licencia Apache. Esta característica ha permitido que cualquier persona acceda al código fuente completo de Android y lo utilice según sus necesidades.

Una de las características distintivas de Android es su capacidad de adaptarse a las necesidades de los fabricantes de dispositivos móviles. Estos fabricantes pueden agregar sus propias extensiones y personalizaciones a Android para diferenciar sus productos en el mercado. Este enfoque modular ha atraído a numerosas empresas, como Motorola y Sony Ericsson, que han aprovechado la flexibilidad de Android para desarrollar sus dispositivos móviles con sistemas operativos personalizados.

Un punto clave de Android es su capacidad para proporcionar un entorno unificado para el desarrollo de aplicaciones móviles. Esto significa que las aplicaciones desarrolladas para Android tienen la capacidad de funcionar en una variedad de dispositivos, siempre que ejecuten el sistema operativo Android. En el competitivo mundo de los smartphones, las aplicaciones móviles son un componente esencial para el éxito, y Android ofrece una plataforma consistente que simplifica el proceso de desarrollo y distribución de aplicaciones. [DiMarzio, 2016]

2.4.3. Patrón MVP

El patrón Modelo-Vista-Presentador es una manera organizada de diseñar la parte de una aplicación que los usuarios ven y con la que interactúan. Se puede comparar con la construcción de una casa, en este caso la casa puede ser una aplicación digital. El MVP es como un plano o un diseño para asegurarse que la casa sea cómoda y funcional para las personas que habitaran en esta.

En este patron se dividen las tareas en tres partes:

- **Modelo:** El modelo es la parte fundamental de una aplicación. Se encarga de almacenar y gestionar los datos necesarios para que la aplicación funcione correctamente. Este almacén de datos puede adoptar diversas formas, como una memoria temporal para guardar información importante o un sistema de comunicación que se utiliza para obtener datos de una base de datos más extensa. En términos más sencillos, podríamos compararlo con el sistema eléctrico de una casa, que proporciona la energía

necesaria para que todo funcione.

- **Vista:** La vista es la parte de la aplicación que los usuarios ven y con la que interactúan directamente. Todos los elementos visibles en la pantalla, como botones, menús y la presentación de información, forman parte de la "Vista". Su función principal es mostrar la información de una manera comprensible y amigable para los usuarios. Se puede pensar en la Vista como la ventana de una casa, a través de la cual observas y experimentas lo que sucede en su interior.
- **Presentador:** El presentador actúa como un intermediario entre el modelo y la vista. Su tarea principal es garantizar que las acciones realizadas por los usuarios en la vista, como hacer clic en un botón o ingresar datos, se traduzcan adecuadamente en instrucciones que el modelo pueda comprender y procesar. En otras palabras, el presentador facilita la comunicación efectiva entre la vista y el modelo para que la aplicación funcione de manera coherente y eficiente. Se puede pensar en el presentador como un traductor que asegura que las dos partes, la vista y el modelo, se entiendan mutuamente y trabajen de manera conjunta.

La idea principal detrás del MVP es hacer que el desarrollo de aplicaciones sea más organizado y fácil de entender. Al separar estas partes, se vuelve más sencillo trabajar en cada una de ellas sin que una interfiera con las demás. Esto hace que la aplicación sea más fácil de mantener y de mejorar con el tiempo, lo que es importante en el mundo digital en constante evolución.[Team, 2023]

El patron MVP tiene ciertas ventajas en comparacion con otros patrones como MVC o MVVM.

1. **Mayor control sobre las vistas:** Una de las principales ventajas del MVP es que el Presentador tiene un control completo sobre la Vista. Esto significa que el Presentador puede modificar la Vista de acuerdo a diferentes estados y funcionalidades sin afectar al Modelo. Esto es especialmente útil en situaciones en las que se requiere una interacción detallada y específica con la interfaz de usuario.
2. **Facilita las pruebas unitarias:** El MVP facilita las pruebas unitarias, ya que la lógica de presentación y la manipulación de la Vista están encapsuladas en el Presentador. Esto permite realizar pruebas de forma más efectiva, ya que se pueden probar las interacciones de la interfaz de usuario de manera aislada, sin necesidad de involucrar al Modelo o a otros componentes.
3. **Desacoplamiento:** El MVP promueve un alto grado de desacoplamiento entre el Modelo, la Vista y el Presentador. Esto significa que cada componente puede evolucionar de manera independiente, lo que facilita la mantenibilidad y la escalabilidad del código. Si es necesario realizar cambios en la interfaz de usuario, se puede lograr sin afectar la lógica del negocio en el Modelo.

4. **Mejora la legibilidad y mantenibilidad del código:** Debido al desacoplamiento y a la separación clara de responsabilidades entre los componentes, el código tiende a ser más legible y más fácil de mantener. Los desarrolladores pueden comprender más fácilmente dónde se encuentra cada parte de la lógica y realizar cambios o mejoras de manera más eficiente.
5. **Adaptable a diferentes plataformas:** El MVP es un patrón que puede aplicarse en una variedad de plataformas y tecnologías, desde aplicaciones web hasta aplicaciones móviles. Esto hace que sea una elección versátil para el desarrollo de software en diferentes contextos.

Es importante mencionar que el uso del patrón MVP puede ser más complejo en proyectos con interfaces de usuario muy ricas o lógica de sincronización extremadamente compleja. En tales casos, pueden ser más adecuados otros patrones de diseño como MVC o MVVM. En cualquier caso la elección del patrón adecuado dependerá en su totalidad de las necesidades específicas de cada proyecto.

En resumen, el Patrón Modelo-Vista-Presentador (MVP) es una metodología valiosa en el desarrollo de aplicaciones, destacando por su capacidad de otorgar un mayor control sobre las vistas, facilitar las pruebas unitarias, promover el desacoplamiento entre los componentes, mejorar la legibilidad del código y ser adaptable a diversas plataformas. Sin embargo, es importante tener en cuenta que en proyectos con interfaces de usuario altamente complejas o lógica de sincronización complicada, otros patrones de diseño pueden ser más adecuados. La elección de utilizar el MVP o no dependerá de las necesidades y características específicas de cada proyecto. En definitiva, el MVP es una herramienta valiosa en el mundo del desarrollo de software, que aporta beneficios significativos en términos de organización y mantenibilidad del código.

2.4.4. Programación orientada a objetos

La Programación Orientada a Objetos (POO) es un paradigma de programación fundamental que ha revolucionado la forma en que concebimos y desarrollamos software. En su núcleo, la POO se basa en la premisa de que el software puede estructurarse y diseñarse de manera más efectiva si lo consideramos como una colección de objetos, donde cada 'objeto' es una entidad autónoma que agrupa tanto datos como funcionalidad.

Cada objeto en la POO representa un elemento del mundo real o una abstracción del mismo. Estos objetos tienen atributos que almacenan datos (también conocidos como propiedades) y métodos (funciones o procedimientos) que definen su comportamiento y cómo interactúan con otros objetos. En esencia, la POO es una herramienta poderosa que nos permite modelar y simular entidades del mundo real y las relaciones entre ellas, creando así una representación computacional de la realidad.

Este enfoque nos permite abordar problemas de programación de una manera más intuitiva y estructurada, ya que podemos dividir un sistema complejo en objetos más pequeños y manejables. Cada objeto se encarga de una parte específica de la funcionalidad, lo que facilita la comprensión y el diseño del software. Además, la POO fomenta la reutilización de código, ya que los objetos se pueden crear a partir de clases existentes y extender su funcionalidad de manera eficiente.

En resumen, la Programación Orientada a Objetos es un enfoque poderoso que nos permite modelar el mundo real en forma de objetos y definir cómo interactúan entre sí. Esto resulta en software más organizado, mantenible y escalable, y ha revolucionado la forma en que abordamos el desarrollo de aplicaciones y sistemas informáticos.

Conceptos claves de la Programación orientada a objetos

- **Encapsulación:** La encapsulación es uno de los conceptos fundamentales de la POO. Implica el ocultamiento de los detalles internos de un objeto y la exposición de una interfaz controlada a través de la cual se puede interactuar con ese objeto. Esto significa que los datos (atributos) de un objeto están protegidos de accesos no autorizados y solo se pueden modificar a través de métodos específicos. La encapsulación garantiza la integridad de los datos y evita cambios no deseados, lo que contribuye a la seguridad y la robustez del código.
- **Abstracción:** La abstracción se refiere a la simplificación de objetos complejos al centrarse en los aspectos esenciales y ocultar los detalles innecesarios. En la POO, la abstracción permite modelar entidades del mundo real de manera efectiva al definir características clave y comportamientos sin tener que considerar todos los detalles complejos. Esto facilita la comprensión y el manejo de objetos, al tiempo que permite a los programadores concentrarse en lo que es relevante para la solución de un problema.
- **Herencia:** La herencia es un principio de la POO que permite la creación de nuevas clases de objetos basadas en clases existentes. Esto fomenta la reutilización de código y la creación de una jerarquía de clases que comparten características comunes. En otras palabras, una nueva clase (llamada 'clase hija' o 'subclase') puede heredar atributos y métodos de una clase existente (llamada 'clase padre' o 'superclase'). La herencia permite crear una estructura de clases organizada y facilita la adaptación de soluciones a nuevos problemas.
- **Polimorfismo:** El polimorfismo es un concepto que permite que diferentes objetos respondan de manera similar a ciertas operaciones. En la POO, esto se logra a través de la implementación de métodos con el mismo nombre pero con comportamientos específicos para cada clase. Esto simplifica la escritura de código, ya que se pueden tratar objetos de diferentes clases de manera uniforme si implementan métodos con el mismo nombre. El polimorfismo permite una mayor flexibilidad y adaptabilidad en el diseño de software.

La programación orientada a objetos tiene ciertas ventajas en comparación a la programación tradicional, estas pueden ser:

- **Modularidad:** La POO fomenta la modularidad al permitir la división del código en módulos reutilizables y auto contenidos, es decir, clases y objetos. Cada módulo representa una entidad o una función específica, lo que facilita el mantenimiento y la comprensión del código. Los cambios y las mejoras se pueden realizar de manera aislada en cada módulo.
- **Reutilización del código:** Uno de los principales beneficios de la POO es la reutilización de código. Los objetos y las clases existentes se pueden utilizar como componentes para crear nuevos objetos y clases. Esto ahorra tiempo y esfuerzo en el desarrollo de nuevas aplicaciones, ya que no es necesario comenzar desde cero.
- **Mantenibilidad:** La estructura organizada y modular de la POO facilita la identificación y corrección de errores. Los cambios y las actualizaciones se pueden realizar de manera más eficiente, ya que no es necesario revisar todo el código. Esto mejora la mantenibilidad a largo plazo del software.
- **Escalabilidad:** La POO permite la incorporación de nuevas funcionalidades sin afectar el funcionamiento existente. Los objetos y las clases se pueden extender o modificar según sea necesario. Esto hace que el software sea escalable y adaptable a medida que los requisitos cambian.

Hoy en día, la programación orientada a objetos se destaca como el paradigma más prevalente en el diseño de aplicaciones y programas informáticos. Esto se debe a sus notables ventajas, especialmente cuando se enfrentan desafíos de programación de gran complejidad. Proporciona una sólida organización de datos y promueve la reutilización de código, lo que a su vez conduce a un ahorro significativo de tiempo a largo plazo.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Descripción general de la solución propuesta.

En vista de los desafíos y problemáticas que hemos identificado hasta el momento en el presente documento, hemos tomado la decisión de abordarlos mediante una solución moderna y tecnológica. En este sentido, nos enfocaremos en la optimización y reducción del uso de los calendarios en sus formatos tradicionales, con la intención de reemplazarlos por la utilización de la aplicación del Calendario Académico de la Universidad Técnica Federico Santa María (UTFSM).

Esta elección se basa en la necesidad de adaptarnos a las tendencias contemporáneas y aprovechar las ventajas que la tecnología nos brinda en el ámbito de la gestión del tiempo y la organización académica. La aplicación del Calendario Académico UTFSM representa un enfoque moderno y eficiente para la planificación y seguimiento de las actividades académicas, proporcionando una serie de beneficios que contribuyen a simplificar y mejorar nuestra experiencia en el entorno universitario.

Entre las ventajas de utilizar esta aplicación se encuentra la posibilidad de acceder a información actualizada de manera instantánea, lo que facilita la consulta de fechas importantes, eventos académicos, exámenes y otras actividades relevantes. Además, la aplicación permite configurar recordatorios y notificaciones, lo que nos ayuda a mantenernos al tanto de nuestras responsabilidades académicas y evitar olvidos o retrasos.

Otro aspecto relevante es la capacidad de personalización que ofrece el Calendario Académico UTFSM, permitiéndonos adaptar la aplicación a nuestras necesidades individuales y preferencias de visualización. Esto contribuye a una mayor eficiencia en la gestión del tiempo y la planificación de nuestras actividades académicas y personales.

En resumen, al optar por esta solución moderna, estamos dando un paso hacia una gestión más eficaz y conveniente de nuestro calendario académico. La tecnología está a nuestro servicio, y utilizarla de manera adecuada puede marcar la diferencia en nuestra productividad y organización en el entorno universitario.

En última instancia, esta propuesta tiene como objetivo cultivar una cultura de aprecio y reconocimiento del Calendario Académico como un pilar fundamental de nuestra organización y planificación. Con ello, esperamos que todos los miembros de nuestra comunidad universitaria aprovechen al máximo esta herramienta valiosa, lo que, a su vez, redundará en un mayor éxito en nuestras actividades académicas y laborales.

3.2. Casos de uso

■ Visualizar el calendario académico:

Descripción: El usuario puede visualizar el calendario académico y otros calendarios relacionados con la universidad en la aplicación móvil de la UTFSM.

Resumen: Los usuarios pueden acceder al calendario en la aplicación y ver eventos, actividades y fechas importantes relacionadas con la universidad.

Actores: Estudiante, profesor, funcionario, externo.

Actor	Sistema
1.- El usuario (Estudiante, profesor, funcionario, externo) presiona el botón de visualizar calendario en la aplicación.	
	2.- El sistema carga la petición del usuario
	3.- El sistema muestra el calendario con los eventos y actividades correspondientes.

Tabla 1: Caso de uso visualizar calendario académico.

Fuente: Elaboración Propia.

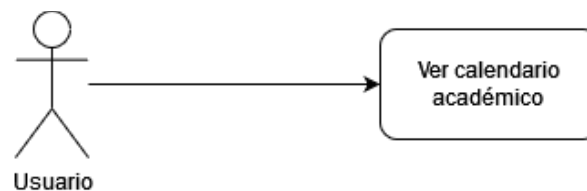


Figura 8: Diagrama visualizar el calendario

Fuente: Elaboración propia.

■ **Registrar actividades de los calendarios:**

Descripción: Los usuarios con los permisos necesarios podrán registrar nuevas actividades en los calendarios existentes.

Resumen: Los usuarios pueden agregar eventos o actividades específicas al calendario, como exámenes, seminarios, reuniones, conferencias, etc.

Actores: Estudiante, Profesor, funcionario.

Actor	Sistema
1.- El usuario selecciona la opción para registrar una nueva actividad	
	2.- El sistema muestra un formulario a rellenar con los campos requeridos como: Día, mes, nombre de la actividad, descripción.
3.- El usuario proporciona los detalles de la actividad y selecciona la opción de registrar actividad	
	4.- El sistema guarda la actividad y la muestra en pantalla.

Tabla 2: Caso de uso registrar actividad.

Fuente: Elaboración Propia.

Excepción: El usuario no completa campos obligatorios.

+El sistema muestra un mensaje indicando que el/los campo/s obligatorio/s no han sido rellenados correctamente y se le solicita al usuario que los complete

Excepción: Fallo en la validación de la fecha.

+Si la fecha ingresada no es válida, el sistema informa al usuario del error y solicita corregir la información.

Excepción: Error al registrar la actividad en la base de datos.

+Si hay un problema al intentar guardar la actividad en la base de datos, el sistema muestra un mensaje de error y sugiere al usuario intentarlo nuevamente.

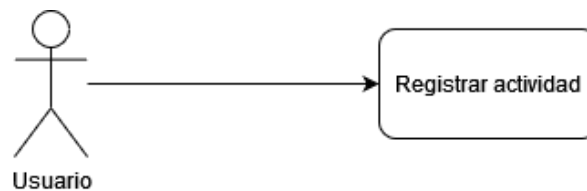


Figura 9: Diagrama registrar actividad.

Fuente: Elaboración propia.

■ **Notificar actividad:**

Descripción: El sistema enviara notificaciones a los usuarios sobre actividades o eventos importantes en el calendario.

Resumen: El sistema envía notificaciones push o mensajes a los usuarios para recordarles eventos próximos o cambios en las actividades del calendario.

Actores: Estudiante, Profesor, funcionario.

Actor	Sistema
	1.- El sistema envía una notificación sobre eventos o actividades marcados como importantes por el usuario.

Tabla 3: Caso de uso notificar actividad.

Fuente: Elaboración Propia.

Excepción: Fallo en el envío de la notificación.

+Si el sistema no puede enviar la notificación push al dispositivo del usuario (por problemas de conectividad, por ejemplo), debe manejar la excepción y registrar el error. Puede intentar reenviar la notificación en un momento posterior.

Excepción: Usuario desactiva las notificaciones.

+Si el usuario ha desactivado las notificaciones push para la aplicación, el sistema debe respetar esta preferencia y no enviar la notificación. Puede registrar este evento para futuras referencias.

Excepción: Error en la recepción de la notificación por parte del usuario.

+Si el usuario no recibe correctamente la notificación (por ejemplo, el dispositivo está apagado), el sistema puede registrar esta excepción para un posterior análisis.

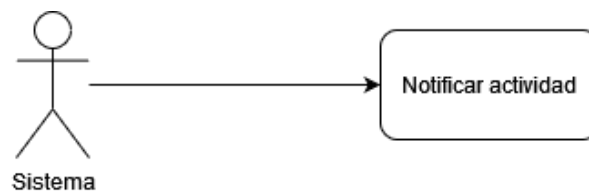


Figura 10: Diagrama caso de uso notificar actividad

Fuente: Elaboración propia.

■ **Registrar perfiles de usuario:**

Descripción: Los usuarios tendrán permitido registrar y gestionar perfiles de diferentes tipos de usuarios en la aplicación.

Resumen: Los usuarios pueden crear perfiles específicos según su rol, como profesor, estudiante, funcionario, externo, etc., lo que les permite acceder a las funciones y características correspondientes.

Actores: Administrador, Profesor, Estudiante, funcionario, Externo, Centro de Estudiantes, Dirección de Sede, Dirección de Adm. y Finanzas, Bienestar Estudiantil.

Actor	Sistema
1.- El usuario crea un perfil de usuario con la información necesaria para cada uno (Nombre, apellido, rut, correo, contraseña, rol.)	
	2.- El sistema muestra en pantalla la confirmación del correo electrónico solicitando ingresar el código que ha sido enviado al correo.
3.- El usuario ingresa el código enviado al correo.	
	4.- El sistema muestra en pantalla que el usuario ha sido registrado exitosamente y lo envía a la pantalla principal de la aplicación.

Tabla 4: Caso de uso registrar perfil de usuario.

Fuente: Elaboración Propia.

Excepción: El usuario no completa los campos obligatorios.

+Si el usuario no completa los campos obligatorios (Nombre Completo, Rut, Correo, Contraseña), se le notificara mediante un mensaje toast y se le solicitara rellenar los campos.

Excepción: Formato incorrecto de datos.

+Si la información ingresada no cumple con el formato esperado (por ejemplo, dirección de correo electrónico inválida), el sistema debe informar al usuario sobre el formato incorrecto y solicitar que corrija los datos.

Excepción: Nombre de usuario o correo electrónico existente.

+Si el nombre de usuario o la dirección de correo electrónico ya están asociados a otro perfil registrado, el sistema debe informar al usuario que debe seleccionar un nombre de usuario o dirección de correo electrónico diferente.

Excepción: Contraseña débil o no cumple con los requisitos.

+Si la contraseña proporcionada no cumple con los requisitos de seguridad (por ejemplo, longitud mínima, uso de caracteres especiales), el sistema debe informar al usuario sobre los requisitos y solicitar una contraseña más segura.

Excepción: Fallo en el registro de la base de datos.

+Si hay un error al intentar registrar la información del usuario en la base de datos (por ejemplo, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que lo vuelva a intentar nuevamente.

Excepción: Problemas con la autenticación en dos pasos.

+Si se utiliza autenticación de dos factores y hay un problema al enviar o validar el código de autenticación, el sistema debe informar al usuario sobre la dificultad y proporcionar una solución o instrucciones alternativas.

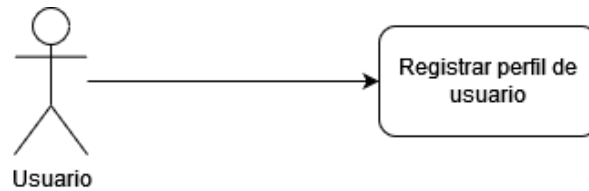


Figura 11: Diagrama caso de uso registrar perfil de usuario
Fuente: Elaboración propia.

- **Registrar departamento:** Descripción: Los usuarios con los permisos correspondientes podrán registrar y administrar los departamentos dentro de la universidad.

Resumen: Los administradores o usuarios autorizados pueden agregar nuevos departamentos a la aplicación, asignar responsables y establecer información relevante para cada departamento.

Actores: Administrador, Dirección de Sede.

Actor	Sistema
1.- El administrador registra el departamento ingresando los datos correspondientes (nombre del departamento, descripción) y asigna responsables.	
	2.- El sistema registra el nuevo departamento en la aplicación.

Tabla 5: Caso de uso registrar departamento.

Fuente: Elaboración Propia.

Excepción: El usuario no ingresa los campos requeridos correctamente.

+Si el usuario no completa los campos obligatorios, se le notificará mediante un mensaje toast y se le solicitará rellenar los campos.

Excepción: Nombre del departamento duplicado.

+Si el nombre del departamento ya existe en el sistema, el sistema debe informar al administrador que debe seleccionar un nombre de departamento único.

Excepción: Fallo en la asignación de responsables.

+En el caso de que exista un problema para asignar responsables, como que el usuario asignado no exista o no tenga los permisos suficientes, se mostrará un mensaje toast en pantalla sobre el error y se sugerirá una solución.

Excepción: Fallo en el registro de la base de datos.

+Si existe un problema al registrar la información en la base de datos, como fallo en la conexión de internet o algún error extra en la base de datos, el sistema le informará al usuario acerca del problema y sugerirá intentarlo nuevamente.

- **Registrar usuario:** Descripción: El usuario que desee podrá registrarse dentro de la aplicación.

Resumen: Los usuarios podrán crear cuentas en caso de Estudiante, profesor, funcionario, externo, etc. Proporcionando la información necesaria para cada tipo de usuario.

Actores: Usuario.

Actor	Sistema
1.- El usuario selecciona el botón de registrar.	
	2.- El sistema envía a la pantalla de registrar usuario con los campos correspondientes (Nombre completo, rut, correo, contraseña).
3.- El usuario rellena cada celda con los datos correspondientes.	
	4.- El sistema registra al usuario en la aplicación mostrando los datos que relleno el usuario anteriormente.

Tabla 6: Caso de uso registrar usuario.

Fuente: Elaboración Propia.

Excepción: Campos vacíos o datos incompletos.

+En el caso que el usuario no complete todos los campos obligatorios o que la información solicitada no cumpla con los requisitos mínimos, se le notificará mediante un mensaje toast al usuario y se le solicitará que complete los campos o en su defecto que ingrese de manera correcta la información solicitada.

Excepción: Correo electrónico o RUT duplicado.

+En el caso que el usuario ingrese un correo electrónico o un RUT que ya esté en uso, se le notificará al usuario y se le solicitará ingresar un correo electrónico o RUT que no esté en uso.

Excepción: Contraseña débil o no cumple con los requisitos.

+Si la contraseña proporcionada no cumple con los requisitos de seguridad (por ejemplo, longitud mínima, uso de caracteres especiales), el sistema debe informar al usuario sobre los requisitos y solicitar una contraseña más segura.

Excepción: Error en el registro en la base de datos.

+Si hay un error al intentar registrar la información del usuario en la base de datos (por ejemplo, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente el registro más tarde.

Excepción: Formato de RUT incorrecto.

+Si el usuario ingresa un RUT no es correcto (Por ejemplo se le solicita que ingrese un RUT con el formato 12345678-9), el sistema debe informar al usuario sobre el RUT y sugerir que ingrese nuevamente el RUT con el formato correcto.

- **Iniciar sesión:** Descripción: Los usuarios previamente registrados podrán iniciar sesión en la aplicación móvil.

Resumen: Los usuarios deben proporcionar sus credenciales de inicio de sesión (correo y contraseña) para acceder a la funcionalidad completa de la aplicación.

Actores: Estudiante, Profesor, funcionario, Externo.

Actor	Sistema
1.- El usuario ingresa sus credenciales de inicio de sesión (correo, contraseña).	
	2.- El sistema verifica las credenciales y autentica al usuario en la aplicación.

Tabla 7: Caso de uso inicio de sesión.

Fuente: Elaboración Propia.

Excepción: Credenciales incorrectas.

+Si el usuario ingresa un correo electrónico o contraseña incorrectos, el sistema debe informar al usuario que las credenciales son inválidas y proporcionar la opción de restablecer la contraseña o intentar nuevamente.

Excepción: Usuario no registrado.

+Si el usuario intenta iniciar sesión con un correo electrónico que no está registrado en la base de datos, el sistema debe informar al usuario que no existe una cuenta asociada con esa dirección de correo electrónico y proporcionar la opción de registrarse.

Excepción: Fallo en la verificación de la base de datos.

+Si hay un error al verificar las credenciales en la base de datos (por ejemplo, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente iniciar sesión nuevamente más tarde.

Excepción: Ataque de fuerza bruta.

+Si hay varios intentos de inicio de sesión fallidos desde una misma cuenta o dirección IP en un corto período de tiempo, el sistema debe aplicar medidas de seguridad, como bloquear temporalmente la cuenta o implementar un sistema de CAPTCHA.

- **Añadir evento al calendario:** Descripción: Los usuarios podrán agregar eventos o actividades específicas a su calendario personal.

Resumen: Los usuarios pueden agregar eventos personalizados, como reuniones personales, recordatorios, fechas importantes, etc., a su calendario en la aplicación.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario selecciona la opción agregar evento.	
	2.- El sistema muestra una pantalla con los campos correspondientes.
3.- El usuario proporciona nombre, y detalles del evento.	
	4.- El sistema verifica si el usuario tiene los permisos necesarios.
	5.- Si tiene los permisos necesarios, se añade el evento al calendario.

Tabla 8: Caso de uso añadir evento al calendario.

Fuente: Elaboración Propia.

Excepción: Campos incorrectos o incompletos.

+Si el usuario no completa todos los campos requeridos o si la información proporcionada es incorrecta, el sistema debe informar al usuario sobre los campos faltantes o incorrectos y solicitar correcciones.

Excepción: Permisos insuficientes.

+Si el usuario no tiene los permisos necesarios para agregar eventos al calendario, el sistema debe informar al usuario sobre la falta de permisos.

Excepción: Conflictos en horario.

+Si el nuevo evento entra en conflicto con eventos existentes en el calendario (por ejemplo, misma fecha y hora), el sistema debe informar al usuario sobre el conflicto y permitirle ajustar la programación.

Excepción: Evento duplicado.

+Si el usuario intenta agregar un evento con el mismo nombre y detalles de uno que ya existe, el sistema debe informar al usuario y sugerir un nombre único o verificar si el evento ya está en el calendario.

Excepción: Error en la verificación de los permisos.

+Si hay un error al verificar los permisos del usuario (por ejemplo, falta de conexión, error en la base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente agregar el evento más tarde.

Excepción: Fallo en la adición del evento en la base de datos.

+Si hay un error al intentar agregar el evento a la base de datos (por ejemplo, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente agregar el evento más tarde.

- **Filtrar evento:** Descripción: Los usuarios tienen permitido filtrar los eventos en el calendario según criterios específicos.

Resumen: Los usuarios pueden aplicar filtros en la aplicación para ver solo ciertos tipos de eventos, fechas específicas o eventos relacionados con áreas o departamentos particulares.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario hace click en el menú para filtrar eventos.	
	2.- El sistema despliega un menú con las opciones de filtros.
3.- El usuario aplica filtros específicos en la aplicación para filtrar eventos.	
	4.- El sistema muestra al usuario solo los eventos que cumplen con los criterios del filtrado.

Tabla 9: Caso de uso filtrar evento.

Fuente: Elaboración Propia.

Excepción: Ningún evento cumple con los filtros.

+Si después de aplicar los filtros no hay eventos que cumplan con los criterios especificados, el sistema debe informar al usuario que no se encontraron eventos y sugerir ajustar los filtros.

Excepción: Fallo en la aplicación de filtros.

+Si hay un error al intentar aplicar los filtros (por ejemplo, error en la lógica de filtrado, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente nuevamente.

Excepción: Error en la presentación de los resultados.

+Si hay un problema al mostrar los resultados filtrados (por ejemplo, fallo en la interfaz de usuario, problemas de rendimiento), el sistema debe informar al usuario y ofrecer soluciones, como reiniciar la aplicación o volver a intentarlo nuevamente.

Excepción: Filtros inválidos o no soportados.

+Si el usuario intenta aplicar filtros que no son válidos o no están soportados por el sistema, el sistema debe informar al usuario sobre la incompatibilidad y proporcionar opciones válidas.

Excepción: Exceso de resultados.

+Si el resultado de la aplicación de filtros genera un conjunto de eventos muy grande, el sistema puede informar al usuario y sugerir criterios de filtrado más específicos para reducir la cantidad de resultados.

- **Buscar evento:** Descripción: Los usuarios pueden hacer búsquedas de eventos o actividades específicas en el calendario.

Resumen: Los usuarios pueden realizar búsquedas utilizando palabras clave o criterios específicos para encontrar eventos o actividades en el calendario que sean de su interés.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario hace click en el boton "Buscar evento".	
	2.- El sistema abre la pantalla para buscar eventos.
3.- El usuario realiza una búsqueda utilizando palabras clave.	
	4.- El sistema muestra en pantalla los eventos que coinciden con la búsqueda realizada.

Tabla 10: Caso de uso buscar evento.

Fuente: Elaboración Propia.

Excepción: Ningún evento coincide con la búsqueda.

+Si después de realizar la búsqueda no se encuentran eventos que coincidan con las palabras clave proporcionadas por el usuario, el sistema debe informar al usuario que no se encontraron eventos y sugerir ajustar los términos de búsqueda.

Excepción: Fallo en la ejecución de la búsqueda.

+Si hay un error al intentar ejecutar la búsqueda (por ejemplo, falta de conexión, error en la lógica de búsqueda, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente la búsqueda nuevamente.

Excepción: Error en la presentación de la búsqueda.

+Si hay un problema al mostrar los resultados de la búsqueda (por ejemplo, fallo en la interfaz de usuario, problemas de rendimiento), el sistema debe informar al usuario y ofrecer soluciones, como recargar la aplicación o volver a intentarlo nuevamente.

Excepción: Búsqueda demasiado amplia o específica.

+Si la búsqueda del usuario genera un conjunto de resultados demasiado grande o demasiado pequeño, el sistema puede informar al usuario y sugerir ajustar los términos de búsqueda para obtener resultados más precisos.

- **Generar informe sobre el calendario:** Descripción: Los usuarios podrán generar informes detallados sobre el calendario y las actividades programadas.

Resumen: Los usuarios pueden generar informes que contengan información relevante sobre los eventos, actividades y fechas importantes en el calendario, que pueden ser utilizados para informes académicos, seguimiento de progreso, planificación, etc.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario selecciona la opción para generar un informe sobre el calendario.	
	2.- El sistema entrega un informe detallado acerca del calendario.

Tabla 11: Caso de uso generar informe.

Fuente: Elaboración Propia.

Excepción: Fallo en la generación del informe.

+Si hay un error al intentar generar el informe (por ejemplo, falta de conexión, error en la lógica de generación, problemas de procesamiento), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente generar el informe nuevamente.

Excepción: Informe vacío o sin datos.

+Si, por alguna razón, el calendario no tiene eventos o datos relevantes para incluir en el informe, el sistema debe informar al usuario que el informe está vacío y proporcionar sugerencias o instrucciones adicionales.

Excepción: Fallo en la entrega del informe.

+Si hay un problema al intentar entregar el informe al usuario (por ejemplo, error en la interfaz de usuario, problemas de envío de archivos), el sistema debe informar al usuario sobre el problema y ofrecer soluciones, como proporcionar un enlace de descarga o enviar el informe por correo electrónico.

- **Compartir evento:** Descripción: Los usuarios tendrán permitido compartir eventos o actividades específicas con otros usuarios o a través de diferentes plataformas.

Resumen: Los usuarios pueden compartir eventos del calendario con otros usuarios de la aplicación o enviar información sobre eventos a través de correos electrónicos, mensajes, redes sociales, etc.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario selecciona un evento para compartir.	
	2.- El sistema proporciona las opciones para compartir el evento.
3.- El usuario elige la plataforma deseada.	
	4.- El sistema envía la información mediante la plataforma elegida por el usuario.

Tabla 12: Caso de uso compartir evento.

Fuente: Elaboración Propia.

Excepción: Fallo al proporcionar opciones para compartir.

+Si hay un error al proporcionar las opciones para compartir el evento (por ejemplo, falta de conexión, error en la lógica de obtención de opciones), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente compartir el evento más tarde.

Excepción: Plataforma de compartir no disponible.

+Si la plataforma elegida por el usuario no está disponible en ese momento (por ejemplo, servicios de redes sociales no disponibles, problemas con el sistema de correo electrónico), el sistema debe informar al usuario y ofrecer alternativas.

Excepción: Fallo en el envío de la información.

+Si hay un error al intentar enviar la información del evento a través de la plataforma elegida (por ejemplo, error en la conexión, problemas con la API de la plataforma), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente compartir el evento nuevamente.

- **Agregar actividad a favoritos** Descripción: Los usuarios tendrán permitido agregar actividades o eventos específicos a su lista de favoritos.

Resumen: Los usuarios pueden marcar eventos como favoritos para tener un acceso rápido y fácil a ellos en su lista de favoritos, sin necesidad de buscarlos nuevamente en el calendario.

Actores: Estudiante, Profesor, Funcionario.

Actor	Sistema
1.- El usuario marca una actividad o evento como favorito.	
	2.- El sistema guarda la actividad o el evento en la lista de favoritos del usuario.

Tabla 13: Caso de uso agregar a favoritos.

Fuente: Elaboración Propia.

Excepción: Fallo al marcar actividad como favorita.

+Si hay un error al intentar marcar la actividad o evento como favorito (por ejemplo, falta de conexión, error en la interfaz de usuario), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente nuevamente.

Excepción: Fallo al guardar actividad en la lista de favoritos.

+Si hay un error al intentar guardar la actividad en la lista de favoritos del usuario (por ejemplo, falta de conexión, error de base de datos), el sistema debe informar al usuario sobre el problema técnico y sugerir que intente agregar la actividad a favoritos nuevamente.

Excepción: Usuario no autenticado.

+Si el usuario no está autenticado y trata de marcar una actividad como favorita, el sistema debe informar al usuario que se requiere iniciar sesión para utilizar la funcionalidad de favoritos y proporcionar instrucciones para iniciar sesión.

3.3. Modelo de datos

El siguiente modelo de datos es una propuesta para la solución abordada sobre la aplicación del calendario académico de la USM.

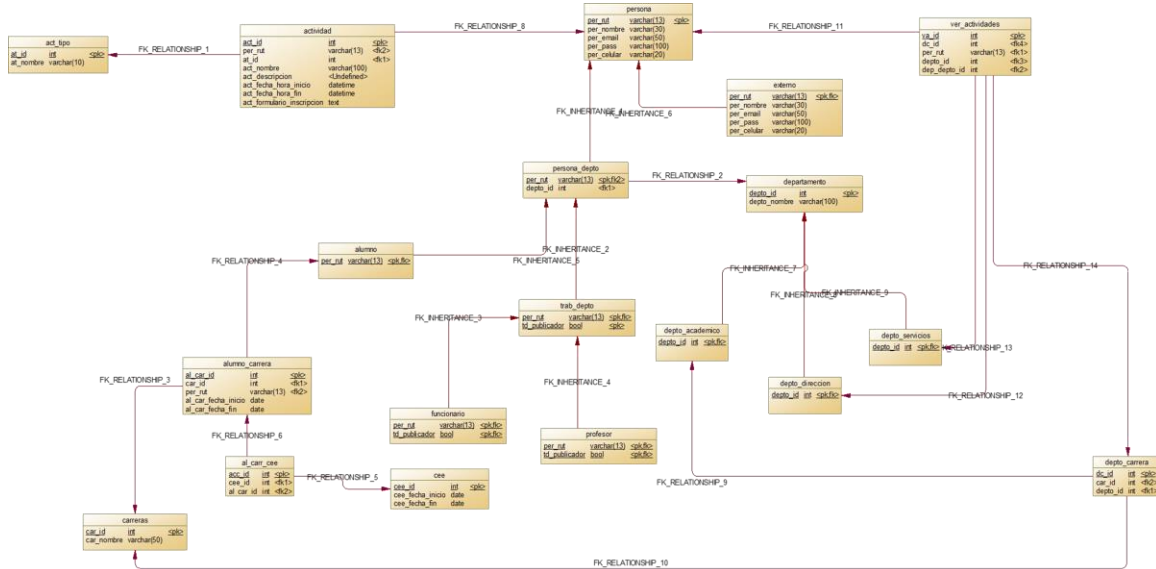


Figura 12: Modelo de datos aplicación calendario académico UTFSM.

Fuente: Elaboración propia.

3.3.1. Explicación de las tablas.

- Nombre de la tabla: Persona.
- Campos de la tabla: per rut, per nombre, per email, per pass, per celular.
- Claves primarias y secundarias: La primary key es per rut, esta almacenará el rut de la persona para que no existan duplicados de la persona.
- Relación con otras tablas: La tabla persona se relaciona con la tabla actividad, persona depto, externo y ver actividades. Para relacionarse con estas tablas se utiliza la primary key 'per rut'.
- Propósito y uso: La tabla persona tiene como finalidad almacenar a los usuarios que se registren en la aplicación.
- Ejemplos de datos: 'per rut: 20.487.047-0', 'per nombre: Diego Lucero', 'per email: diego.luceroq@usm.cl', 'per pass: D120000605', 'per celular: +56976730200'.
- Actualizaciones y mantenimiento: En caso de que el usuario esté inactivo después de pedir una confirmación de uso de la app, el usuario se eliminará automáticamente.

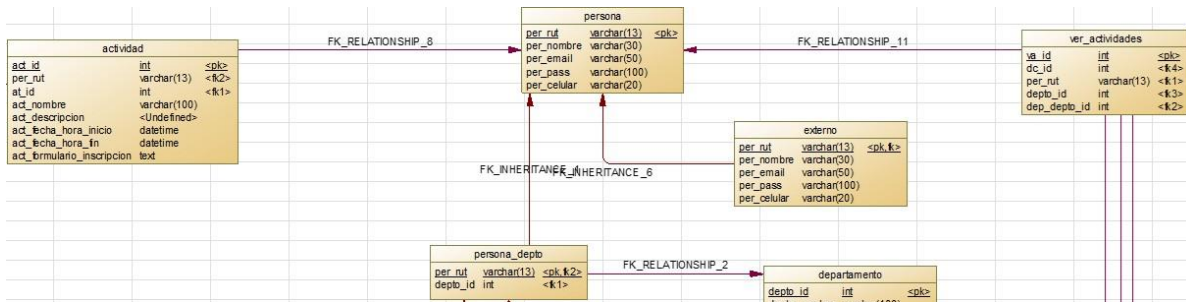


Figura 13: Relación de tabla persona.

Fuente: Elaboración propia.

- Nombre de la tabla: Actividad.
- Campos de la tabla: act id, per rut, at id, act nombre, act descripción, act fecha hora inicio, act fecha hora fin, act formulario inscripción.
- Claves primarias y secundarias: La primary key de la tabla actividad es act id, esto es para evitar duplicados de las actividades. Las fk son per rut, este para asociar el rut de la persona que ingresará la actividad, la marcará como favorito, etc. y at id para almacenar la id de la tabla act tipo.
- Relación con otras tablas: La tabla actividad tiene relación directa con act tipo mediante la fk de 'at id', y con persona que tiene relación mediante la pk de 'per rut'.
- Propósito y uso: La tabla actividad tiene como finalidad almacenar las actividades que tiene el calendario académico.
- Ejemplos de datos: 'act id: 1', 'per rut: 20.487.047-0', 'at id: 1', act nombre: 'Ceremonia conmemoración del fallecimiento de Don Federico Santa Maria', act descripción: 'Se conmemora el fallecimiento de Don Federico Santa Maria', act fecha hora inicio: '20/12/2023', act fecha hora fin: 'Unknown', act formulario inscripción: ''.
- Actualizaciones y mantenimiento: En caso de que la actividad se retrase por problemas, el encargado de la actividad podrá modificar la hora de inicio o en su defecto eliminarla en caso de que se cancele por algún motivo.



Figura 14: Relación de la tabla actividad

Fuente: Elaboración propia.

- Nombre de la tabla: Actividad Tipo.
- Campos de la tabla: at id, at nombre.
- Claves primarias y secundarias: La pk de la tabla act tipo es at id la cual almacena la id de cada tipo de actividad.
- Relación con otras tablas: La tabla act tipo se relaciona directamente con la tabla actividad, mediante la pk at id.
- Propósito y uso: La tabla actividad tipo tiene como finalidad almacenar el tipo de actividad que se intenta crear.
- Ejemplos de datos: 'at id: 1', 'at nombre: Ceremonia'.
- Actualizaciones y mantenimiento: En caso de que la tabla actividad tipo tenga algún nombre de tipo de actividad ya ingresado se deberá reingresar un nombre nuevo.

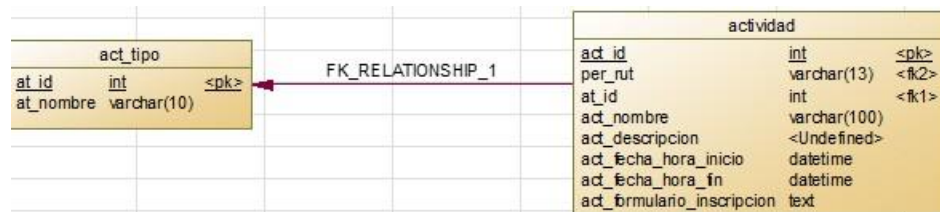


Figura 15: Relación de la tabla act tipo
Fuente: Elaboración propia.

- Nombre de la tabla: Persona depto.
- Campos de la tabla: per rut, depto id.
- Claves primarias y secundarias: La pk de la tabla persona detpo es per rut que al mismo tiempo es una fk, esta se relaciona con la tabla persona para heredar directamente el rut de la persona, la otra fk es depto id, este toma la id de la tabla departamento para saber a que departamento pertenece la persona.
- Relación con otras tablas: La tabla persona depto se relaciona directamente con la tabla persona, de la cual hereda el rut, junto a la tabla departamento la cual extrae directamente la id del departamento al que pertenece la persona.
- Propósito y uso: La tabla persona depto tiene como finalidad almacenar a la persona que pertenece a cierto departamento.
- Ejemplos de datos: 'per rut: 20.487.047-0', 'depto id: 1'.
- Actualizaciones y mantenimiento: En el caso de que el rut no pertenezca a ningún departamento se deberá revisar o simplemente borrar el usuario.



Figura 16: Relación de la tabla persona depto.

Fuente: Elaboración propia.

- Nombre de la tabla: Externo.
- Campos de la tabla: per rut, per nombre, per email, per pass, per celular.
- Claves primarias y secundarias: La pk de esta tabla es per rut que hereda directamente desde la tabla persona el rut de la persona que se registrará como externo.
- Relación con otras tablas: La tabla externo se relaciona directamente con la tabla persona.
- Propósito y uso: La tabla externo tiene como finalidad almacenar todo usuario que sea externo a la Universidad.
- Ejemplos de datos: 'per rut: 19.546.245-7', 'per nombre: Fernanda Muñoz', 'per email: fermun@gmail.com', 'per pass: fermun123456', 'per celular: +56945366672'.
- Actualizaciones y mantenimiento: En el caso de que el usuario externo permanezca mucho tiempo inactivo (por ejemplo sin entrar a la aplicación) será eliminado instantáneamente de la base de datos.

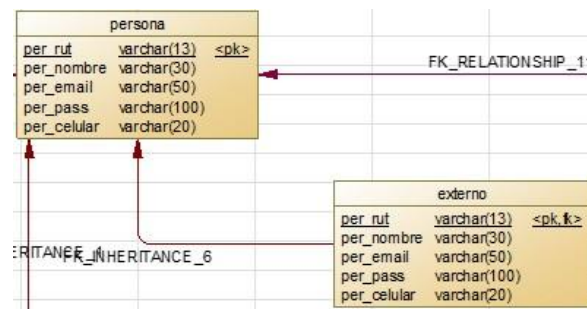


Figura 17: Relación de la tabla externo

Fuente: Elaboración propia.

- Nombre de la tabla: Ver actividades.
- Campos de la tabla: va id, dc id, per rut, depto id, dep depto id.

- Claves primarias y secundarias: La pk de la tabla ver actividades es la id, seguido de la primera fk que es dc id el cual hereda la id del departamento de carrera, siguiendo con el rut de la persona que hereda directamente de la tabla persona para almacenar la persona que visualiza la actividad, la tercera fk es depto id que hereda la id del departamento, la cuarta fk es dep depto id que hereda la id directamente de las tablas depto direccion y depto servicios.
- Relación con otras tablas: Ver actividades se relaciona con 4 tablas, con persona, depto direccion, depto servicios, depto carrera.
- Propósito y uso: La tabla ver actividades, como su nombre indica, es para llevar un registro de las actividades visualizadas por el usuario.
- Ejemplos de datos: 'va id: 1', 'dc id: 1', 'per rut: 20.487.047-0', 'depto id: 1', 'dep depto id: 1'.
- Actualizaciones y mantenimiento: Las actividades se iran renovando cuando el calendario academico se actualce anualmente.

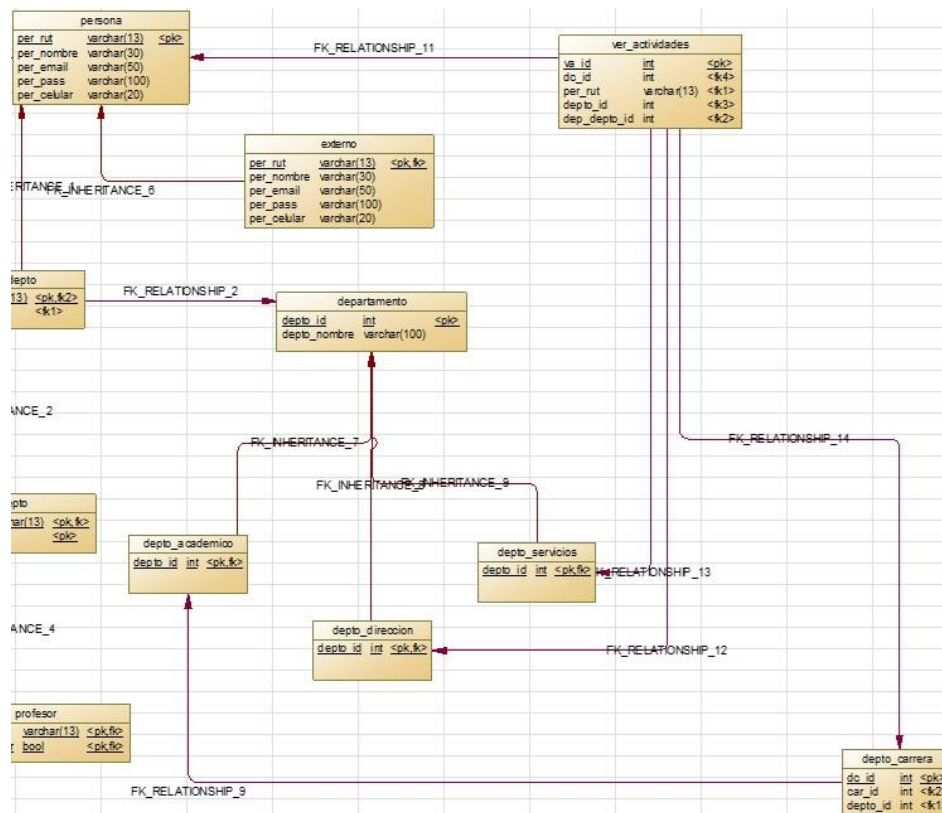


Figura 18: Relacion de la tabla ver actividades
Fuente: Elaboracion propia.

- Nombre de la tabla: Departamento.

- Campos de la tabla: depto id, depto nombre.
- Claves primarias y secundarias: La pk de la tabla departamento es depto id, la cual almacena el id del departamento para que cada departamento tenga su id propio.
- Relación con otras tablas: Ver actividades se relaciona con 4 tablas, con persona dep- to, la cual hereda el depto id para saber a que departamento pertenece. Aparte se relaciona con depto academico, depto direccion, depto servicios.
- Propósito y uso: La tabla departamento tiene como finalidad almacenar todos los de- partamentos con su id correspondiente y sus nombres.
- Ejemplos de datos: 'depto id: 1', 'depto nombre: Departamento de Electronica e Infor- matica'.
- Actualizaciones y mantenimiento: La tabla departamento se debe ir actualizando a medida que vayan apareciendo o eliminando departamentos de la Universidad.

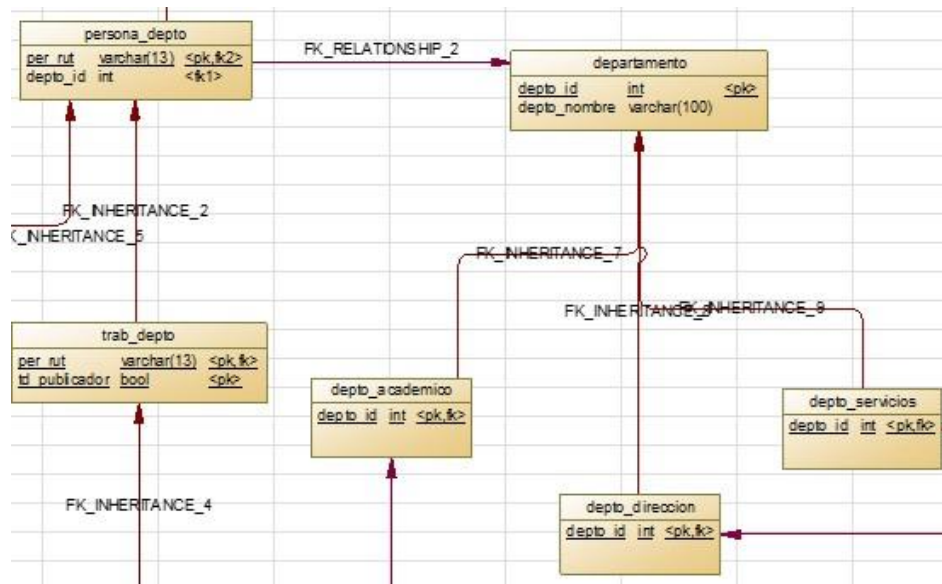


Figura 19: Relacion de la tabla Departamento
Fuente: Elaboracion propia.

- Nombre de la tabla: Trab depto.
- Campos de la tabla: per rut, td publicador.
- Claves primarias y secundarias: La pk de trab depto que al mismo tiempo es una fk es per rut, que hereda de persona depto para saber si el trabajador pertenece a una persona del departamento, la principal pk de esta tabla es td publicador que en este caso es un boolean para saber si tiene permitido publicar un evento o no.

- Relación con otras tablas: La tabla trab depto se relaciona directamente con persona depto, funcionario y profesor.
- Propósito y uso: La tabla trab depto tiene como finalidad almacenar a los trabajadores del departamento correspondiente, viendo si tiene permisos para publicar una actividad o no.
- Ejemplos de datos: 'per rut: 20.487.047-0', 'td publicador: True'.
- Actualizaciones y mantenimiento: La tabla trab depto se debe ir actualizando constantemente para agregar o remover a los trabajadores que tienen permisos de publicación.

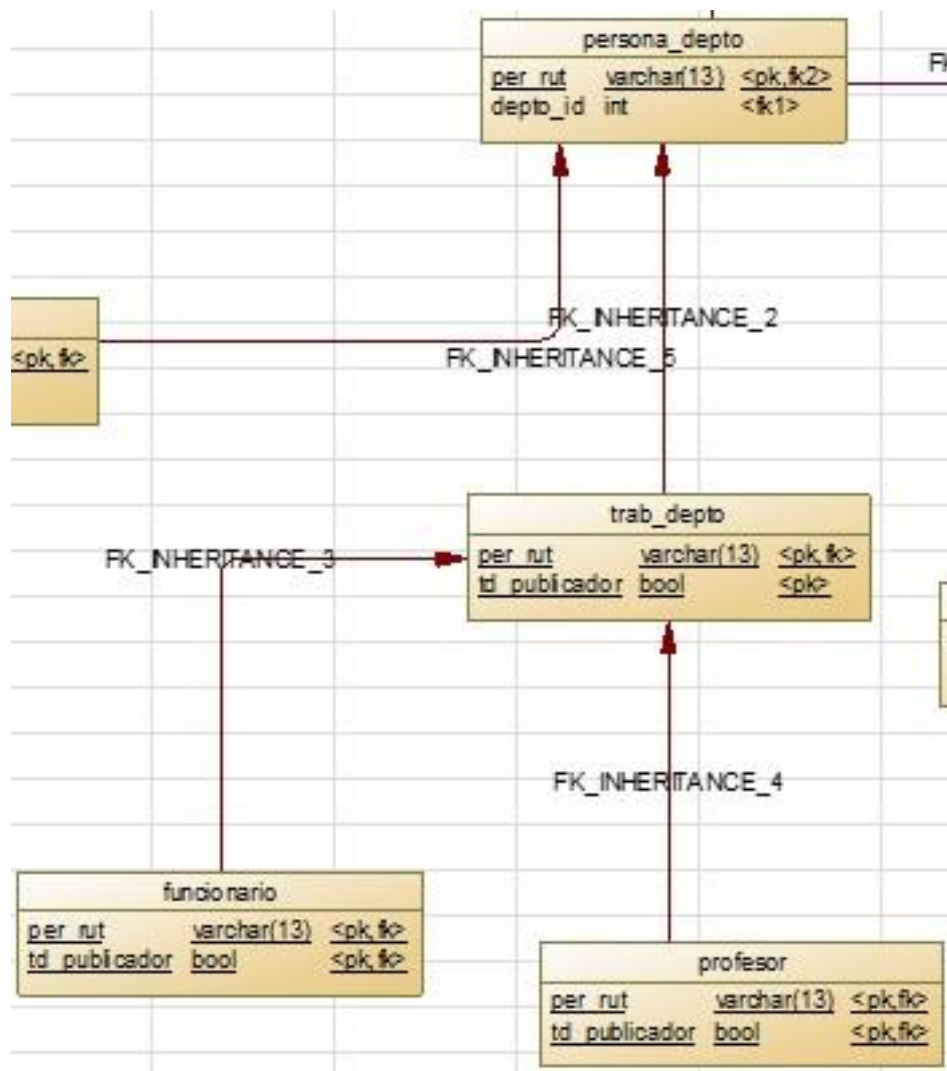


Figura 20: Relacion de la tabla trab depto.
Fuente: Elaboracion propia.

- Nombre de la tabla: Funcionario.

- Campos de la tabla: per rut, td publicador.
- Claves primarias y secundarias: Funciona de la misma manera que trab depto.
- Relación con otras tablas: La tabla funcionario se relaciona directamente con trab depto.
- Propósito y uso: La tabla funcionario tiene como finalidad almacenar a todos los funcionarios de la USM.
- Ejemplos de datos: 'per rut: 18.545.265-9', 'td publicador: True'.
- Actualizaciones y mantenimiento: La tabla funcionario se debe ir modificando constantemente ya que siempre hay funcionarios nuevos o se van.

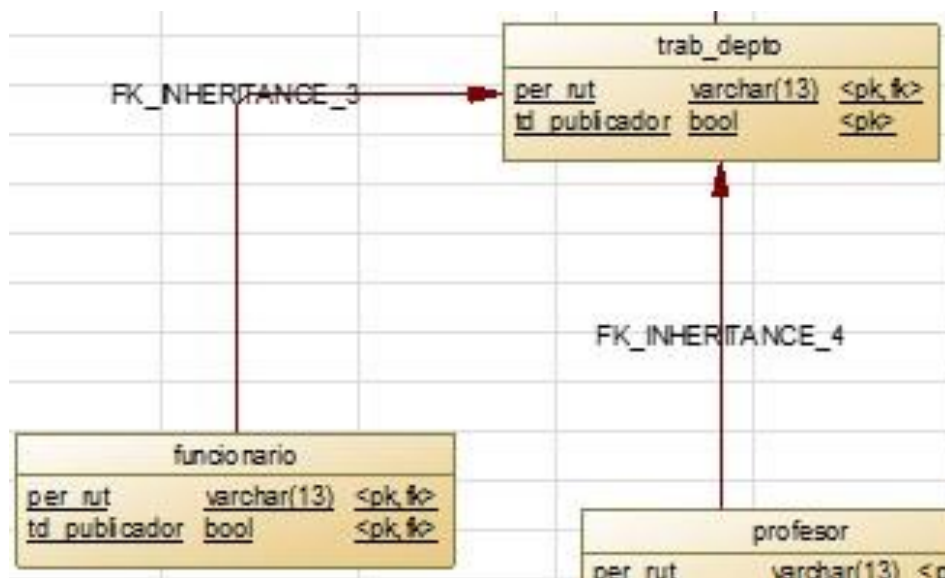


Figura 21: Relacion de la tabla funcionario.
Fuente: Elaboracion propia.

- Nombre de la tabla: Profesor.
- Campos de la tabla: per rut, td publicador.
- Claves primarias y secundarias: Funciona de la misma manera que trab depto y funcionario.
- Relación con otras tablas: La tabla profesor se relaciona directamente con trab depto.
- Propósito y uso: La tabla profesor tiene como finalidad almacenar a todos los profesores de la USM.
- Ejemplos de datos: 'per rut: 17.766.341-4', 'td publicador: False'.

- Actualizaciones y mantenimiento: La tabla profesor se debe ir modificando constantemente en caso de que un profesor nuevo se una o en su defecto removerlo.

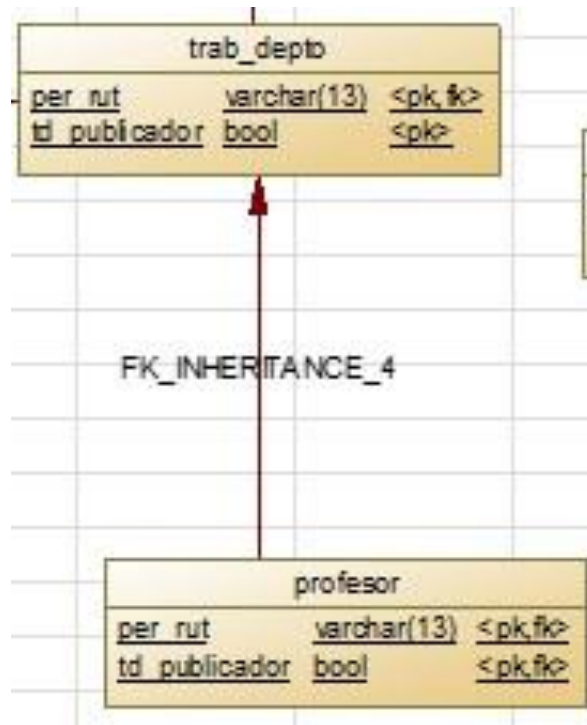


Figura 22: Relacion de la tabla profesor.
Fuente: Elaboracion propia.

- Nombre de la tabla: Alumno.
- Campos de la tabla: per rut.
- Claves primarias y secundarias: La pk de alumno es solamente el rut, esto con el fin de saber el rut de los alumnos.
- Relación con otras tablas: La tabla alumno se relaciona directamente con persona dep- to y alumno carrera.
- Propósito y uso: La tabla alumno tiene como finalidad almacenar los alumnos que ingresan a la USM.
- Ejemplos de datos: 'per rut: 22.015.256-6'.
- Actualizaciones y mantenimiento: La tabla alumno se debe ir modificando constantemente debido al ingreso de alumnos y a la salida de estos mismos.

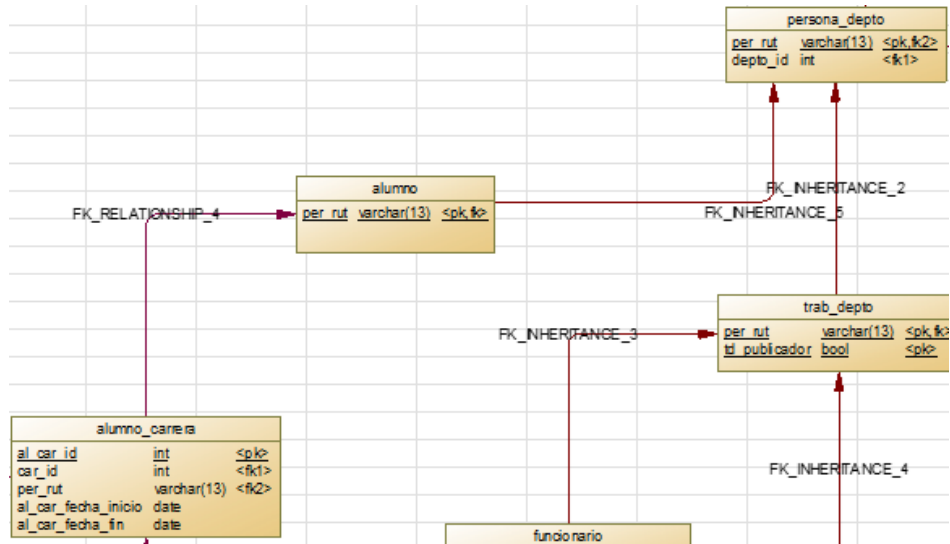


Figura 23: Relacion de la tabla Alumno.

Fuente: Elaboracion propia.

- Nombre de la tabla: Alumno carrera.
- Campos de la tabla: al car id, car id, per rut, al car fecha inicio, al car fecha fin.
- Claves primarias y secundarias: La pk de alumno car es 'al car id', que tiene como finalidad darle un id a un alumno en cierta carrera. Cuenta con dos fk, 'car id' para saber la id de la carrera a la que pertenece y 'per rut' para saber el rut de la persona que esta en esa carrera.
- Relación con otras tablas: La tabla alumno carrera se relaciona directamente con la tabla alumno para saber que alumno corresponde cada carrera, junto con al carr cee para saber si el alumno pertenece al centro de estudiantes.
- Propósito y uso: La tabla alumno carrera tiene como finalidad conocer a que carrera pertenece cada alumno ingresado en la aplicacion.
- Ejemplos de datos: 'al car id: 1', 'car id: 1', 'per rut: 20.947.972-9', 'car fecha inicio: 10/04/2024', 'car fecha fin: Unknown'.
- Actualizaciones y mantenimiento: La tabla alumno carrera se ira actualizando a medida que ingresen nuevos estudiantes a las carreras o en su defecto egresen.

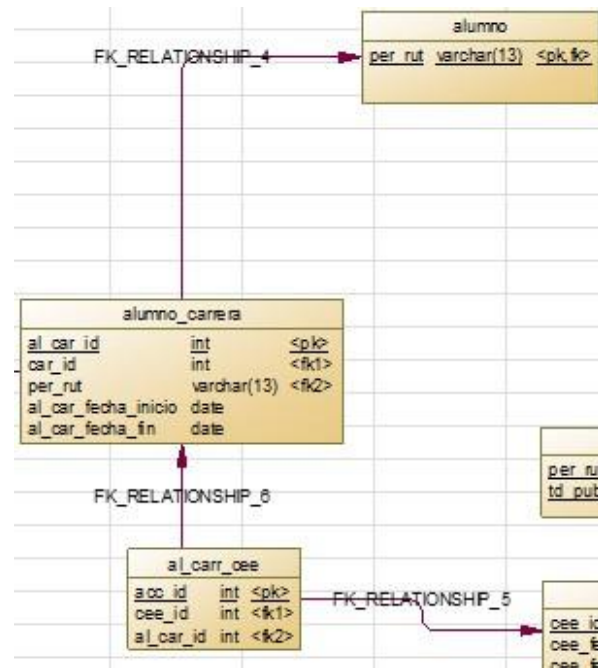


Figura 24: Relacion de la tabla Alumno carrera.
Fuente: Elaboracion propia.

- Nombre de la tabla: al carr cee.
- Campos de la tabla: acc id, cee id, al car id.
- Claves primarias y secundarias: La pk de esta tabla es acc id, que le da una id unica al alumno de cierta carrera que pertenece al centro de estudiante. Cuenta con dos fk, estas son cee id, que reconoce el id del centro de estudiante al que pertenece y al car id para saber el id de la carrera a la que pertenece el estudiante.
- Relación con otras tablas: Se relaciona directamente con alumno carrera y cee.
- Propósito y uso: La finalidad de esta tabla es saber que alumno pertenece a cierto centro de estudiantes.
- Ejemplos de datos: 'acc id: 1', 'cee id: 1', 'al car id: 1'.
- Actualizaciones y mantenimiento: La tabla se ira actualizando a medida que ingresen nuevos estudiantes al centro de estudiantes o en su defecto salgan.

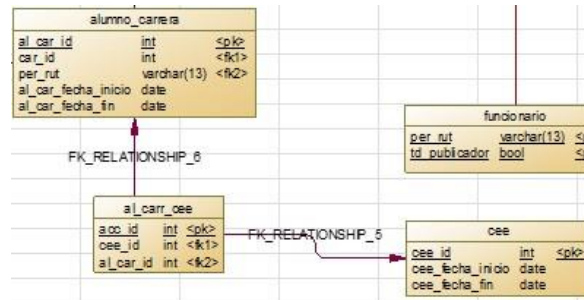


Figura 25: Relacion de la tabla al carr cee.
Fuente: Elaboracion propia.

- Nombre de la tabla: cee
- Campos de la tabla: cee id, cee fecha inicio, cee fecha fin.
- Claves primarias y secundarias: La pk de cee es cee id que tiene como finalidad asignarle una id al centro de estudiantes.
- Relación con otras tablas: Se relaciona directamente con al carr cee.
- Propósito y uso: Tiene como finalidad almacenar los centros de estudiantes actualmente vigentes.
- Ejemplos de datos: 'cee id: 1', 'cee fecha inicio: 14/05/2023', 'cee fecha fin: 01/12/2023'.
- Actualizaciones y mantenimiento: La tabla se ira actualizando a medida que los centros de estudiantes terminen su cargo.



Figura 26: Relaciond e la tabla cee.
Fuente: Elaboracion propia.

- Nombre de la tabla: Carreras.
- Campos de la tabla: car id, car nombre.
- Claves primarias y secundarias: La pk de la tabla carreras es car id, con el fin de asignarle una id a las carreras existentes en la Universidad.
- Relación con otras tablas: La tabla carreras se relaciona directamente con alumno carrera y depto carrera.

- Propósito y uso: La tabla carreras tiene la finalidad de almacenar todas las carreras existentes en la USM.
- Ejemplos de datos: 'car id: 1', 'car nombre: Ingeniería en Informática'.
- Actualizaciones y mantenimiento: La tabla se debe mantener con sus id's correspondientes para no generar conflictos en la base de datos.

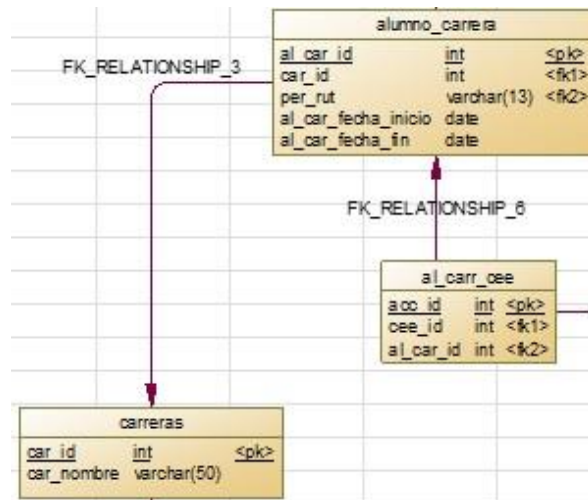


Figura 27: Relacion de la tabla carreras.
Fuente: Elaboracion propia.

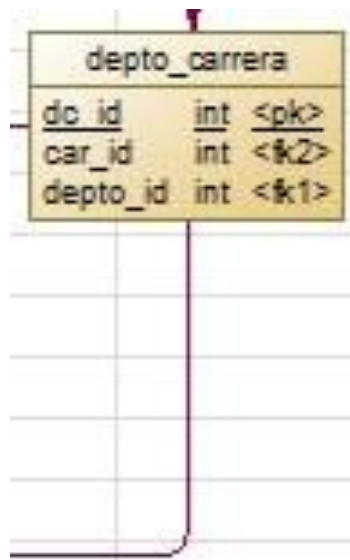


Figura 28: Relacion de la tabla carreras.
Fuente: Elaboracion propia.

- Nombre de la tabla: depto academico.

- Campos de la tabla: depto id.
- Claves primarias y secundarias: La pk de depto academico es depto id, esto con la finalidad de asignarle una id unica a un departamento academico.
- Relación con otras tablas: Se relaciona directamente con departamento y depto carrera.
- Propósito y uso: La tabla depto academico tiene la finalidad de almacenar los id's de los departamentos academicos.
- Ejemplos de datos: 'depto id: 1'.
- Actualizaciones y mantenimiento: Los departamentos deben ser actualizados manualmente por el encargado del sistema de la aplicacion.

- Nombre de la tabla: depto direccion.
- Campos de la tabla: depto id.
- Claves primarias y secundarias: La pk de depto direccion es depto id, esto con la finalidad de asignarle una id unica a un departamento de direccion.
- Relación con otras tablas: Se relaciona directamente con departamento y ver actividades.
- Propósito y uso: La tabla depto direccion tiene como finalidad almacenar las id's de los departamentos de direccion existentes.
- Ejemplos de datos: 'depto id: 3'.
- Actualizaciones y mantenimiento: Los departamentos deben ser actualizados manualmente por el encargado del sistema de la aplicacion.

- Nombre de la tabla: depto servicios.
- Campos de la tabla: depto id.
- Claves primarias y secundarias: La pk de depto servicios es depto id, esto con la finalidad de asignarle una id unica a un departamento de servicios.
- Relación con otras tablas: Se relaciona directamente con departamento y ver actividades.
- Propósito y uso: La tabla depto servicios tiene la finalidad de almacenar los id's de los departamentos de servicios existentes.
- Ejemplos de datos: 'depto id: 9'.

- Actualizaciones y mantenimiento: Los departamentos deben ser actualizados manualmente por el encargado del sistema de la aplicación.

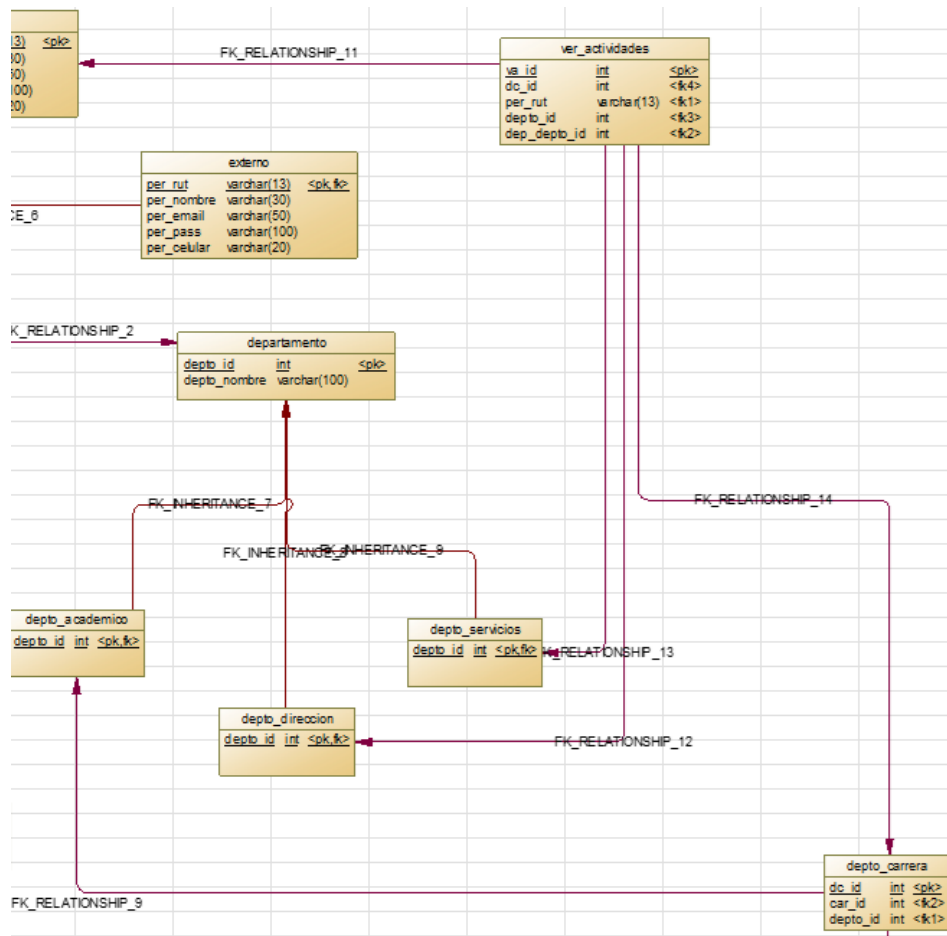


Figura 29: Relacion de las tablas depto academico, depto direccion y depto servicios.
Fuente: Elaboracion propia.

3.4. Integración de Tecnologías.

Aquí se hablara acerca de todas las tecnologías que se usaran al momento de desarrollar la aplicación.

3.4.1. Introducción a las tecnologías.

Razones para la eleccion de Kotlin: Kotlin ha sido elegido como el lenguaje principal de programación debido a varias razones fundamentales. En primer lugar, como parte integral de

nuestro curso de aplicaciones móviles, Kotlin ofrece una sintaxis clara y concisa que facilita el aprendizaje y la implementación efectiva de soluciones móviles. Su interoperabilidad con Java, un lenguaje ampliamente utilizado en el desarrollo de aplicaciones Android, proporciona una transición suave y una mayor flexibilidad para aprovechar recursos existentes.

Además, Kotlin destaca por sus características modernas, como la nulabilidad segura y funciones de orden superior, que no solo mejoran la calidad del código sino que también simplifican el manejo de complejidades inherentes al desarrollo de aplicaciones móviles.

Uso del patrón MVP: Para estructurar nuestra aplicación de manera eficiente y sostenible, se ha adoptado el patrón de diseño Modelo-Vista-Presentador (MVP). Este modelo proporciona una separación clara de las responsabilidades, mejorando la mantenibilidad y la comprensión del código. La elección de MVP se alinea con las mejores prácticas de desarrollo de software, permitiendo una evolución más eficiente de la aplicación a medida que crece en complejidad.

Consideraciones sobre el desarrollo en Mac: En una primera instancia, la aplicación se centrará en usuarios de Android. Esta decisión está motivada por la dificultad de conseguir los medios necesarios para el desarrollo en un entorno Mac, un desafío que se ha enfrentado en este contexto académico. Sin embargo, esta elección no limitará futuras expansiones o la posibilidad de adaptar la aplicación a otras plataformas.

Al seleccionar Kotlin como nuestro lenguaje de programación principal y adoptar el modelo MVP, buscamos no solo cumplir con los requisitos del curso, sino también sentar las bases para una aplicación robusta, escalable y fácil de mantener.

3.4.2. Detalles de Kotlin y Android Studio.

Kotlin como Lenguaje de programación:

1. Interoperabilidad con Java:

- Descripción: Kotlin ha sido seleccionado por su excelente interoperabilidad con Java, permitiendo la integración suave con bibliotecas y código Java existente en el ecosistema de desarrollo Android.
- Facilita la transición y la utilización de recursos previos desarrollados en Java, mejorando la eficiencia del desarrollo.

2. Sintaxis concisa y legible:

- Descripción: La sintaxis concisa y legible de Kotlin simplifica el desarrollo y mejora la comprensión del código.
- Aprovechamiento: Facilita el aprendizaje para los estudiantes y optimiza la eficiencia en la implementación de soluciones móviles.

3. Características modernas:

- Descripción: Características como la nulabilidad segura y las funciones de orden superior mejoran la calidad del código y simplifican el manejo de casos complejos.
- Aprovechamiento: Contribuyen a la robustez y mantenibilidad del código, especialmente en el contexto de desarrollo de aplicaciones móviles.

Android Studio como entorno de desarrollo:

1. Herramientas integradas:

- Descripción: Android Studio proporciona herramientas integradas específicas para el desarrollo de aplicaciones Android, como el diseñador de interfaces gráficas y el depurador.
- Aprovechamiento: Facilita la creación y modificación de interfaces de usuario, así como la identificación y corrección de errores durante el desarrollo.

2. Emuladores y dispositivos virtuales:

- Descripción: Android Studio incluye emuladores y dispositivos virtuales que permiten probar y depurar aplicaciones en diferentes versiones de Android y tamaños de pantalla.
- Aprovechamiento: Facilita la verificación del rendimiento y la apariencia de la aplicación en diversos entornos antes de la implementación en dispositivos reales.

3. Integración con servicios Google:

- Descripción: Android Studio se integra directamente con servicios de Google, como Firebase, simplificando la implementación de funciones como la autenticación, la base de datos en tiempo real y las notificaciones en la aplicación.
- Aprovechamiento: Permite aprovechar servicios en la nube para mejorar la funcionalidad y la experiencia del usuario de la aplicación.

3.4.3. Patrón de diseño MVP.

El patrón de diseño Modelo-Vista-Presentador (MVP) es una arquitectura que separa las responsabilidades de presentación y lógica de negocio en una aplicación. Este patrón facilita la creación de aplicaciones más estructuradas y mantenibles al dividir la aplicación en tres componentes principales: el Modelo (Model), la Vista (View) y el Presentador (Presenter).

El uso del patrón MVP presenta varios beneficios, siendo los más destacados:

1. Separación de responsabilidades:

- MVP separa claramente las responsabilidades de la lógica de negocio (Presentador) y la representación visual (Vista), facilitando la comprensión y el mantenimiento del código.

2. Facilita las pruebas unitarias:

- La separación de las capas permite realizar pruebas unitarias de manera más efectiva, ya que la lógica de presentación se encuentra en el Presentador y puede ser probada de manera independiente de la Vista.

3. Mejora la mantenibilidad:

- La estructura clara del patrón MVP facilita la identificación y corrección de errores, así como la incorporación de nuevas características sin afectar otras partes del sistema.

Responsabilidades de cada componente:

1. Modelo:

- Descripción: Representa los datos y la lógica de negocio de la aplicación.
- Responsabilidades
 - Gestiona la recuperación y persistencia de datos.
 - Aplica la lógica de negocio de la aplicación.
 - Notifica al Presentador sobre cambios en los datos.

2. Vista:

- Descripción: Representa la interfaz de usuario y muestra los datos al usuario.
- Responsabilidades:
 - Muestra la información al usuario.
 - Captura eventos del usuario (interacciones con la interfaz).
 - Comunica las acciones del usuario al Presentador.

3. Presentador:

- Descripción: Actúa como intermediario entre el Modelo y la Vista.
- Responsabilidades:
 - Recupera datos del Modelo y los presenta a la Vista.
 - Maneja las interacciones del usuario desde la Vista y realiza las operaciones correspondientes en el Modelo.
 - Contiene la lógica de presentación y toma decisiones basadas en la información del Modelo y las acciones del usuario.

3.4.4. Consideraciones de rendimiento y eficiencia.

Las tecnologías seleccionadas, Kotlin como lenguaje de programación y Android Studio como entorno de desarrollo, pueden contribuir significativamente a la eficiencia y al rendimiento de la aplicación del calendario académico.

1. Kotlin:

- **Concisión del Código:** La sintaxis concisa de Kotlin permite escribir menos código en comparación con Java, lo que conduce a una mayor eficiencia de desarrollo y una menor probabilidad de errores.
- **Interoperabilidad con Java:** Kotlin se integra perfectamente con el código Java existente, lo que facilita la reutilización de bibliotecas Java y la incorporación de código Kotlin en proyectos Android existentes.
- **Null Safety:** La nulabilidad segura de Kotlin ayuda a reducir errores relacionados con null, mejorando la estabilidad y eficiencia del código.
- **Soporte de Funciones de Orden Superior:** La capacidad de Kotlin para utilizar funciones de orden superior facilita la escritura de código más modular y reutilizable, lo que contribuye a un diseño más eficiente.

2. Android Studio:

- **Herramientas de Optimización del Código:** Android Studio proporciona herramientas integradas, como el Lint, que identifican y sugieren mejoras en el código, contribuyendo a un código más eficiente y de alto rendimiento.
- **Emuladores y Dispositivos Virtuales:** Los emuladores y dispositivos virtuales de Android Studio permiten probar la aplicación en una variedad de configuraciones, facilitando la identificación de posibles cuellos de botella de rendimiento.
- **Integración con Perfiles de Rendimiento:** Android Studio proporciona herramientas para realizar perfiles de rendimiento en la aplicación, lo que permite identificar y abordar áreas que podrían afectar negativamente al rendimiento.
- **Soporte para Desarrollo Nativo de Android:** Android Studio está diseñado específicamente para el desarrollo de aplicaciones Android, lo que significa que proporciona un entorno optimizado y herramientas específicas para las necesidades de desarrollo en esta plataforma.

3. Enfoque en desarrollo nativo de Android:

- **Acceso a Funcionalidades Nativas:** Utilizar Kotlin y Android Studio permite un acceso más directo a las funcionalidades nativas de Android, lo que puede traducirse en un rendimiento más eficiente y una mejor integración con el sistema operativo.

- Optimización para Android Runtime (ART): Kotlin está optimizado para Android Runtime (ART), lo que significa que puede ejecutar código más eficientemente en dispositivos Android, contribuyendo a un mejor rendimiento.

3.4.5. Manejo de errores y excepciones.

1. Kotlin:

a) Nulabilidad segura:

- Facilitación del Manejo de Errores: Kotlin introduce el concepto de nulabilidad segura, lo que significa que el sistema de tipos de Kotlin distingue entre referencias nulas y no nulas. Esto reduce significativamente las posibilidades de errores relacionados con valores nulos, ya que el compilador advierte sobre posibles riesgos.

b) Expresiones de excepción mejoradas:

- Facilitación del Manejo de Excepciones: Kotlin simplifica la sintaxis para trabajar con excepciones. Se pueden utilizar expresiones más concisas para manejar excepciones, lo que mejora la legibilidad y facilita la gestión de casos excepcionales.

c) Funciones de orden superior:

- Gestión Delegada de Errores: Las funciones de orden superior en Kotlin permiten la delegación de la gestión de errores. Puedes pasar funciones como parámetros, facilitando la encapsulación y manejo de errores en funciones específicas.

2. Android Studio:

a) Lint y análisis estadístico:

- Identificación de Problemas en Tiempo de Desarrollo: Android Studio incluye herramientas como el Lint, que realiza análisis estático del código en tiempo de desarrollo. Esto ayuda a identificar posibles problemas y errores antes de la ejecución, mejorando la calidad del código y reduciendo la probabilidad de errores en tiempo de ejecución.

b) Depurador Integrado:

- Facilitación del Diagnóstico de Errores: Android Studio proporciona un depurador integrado que facilita la identificación y diagnóstico de errores durante el desarrollo. Esto agiliza el proceso de corrección de errores y mejora la eficiencia del desarrollo.

c) Asistentes y Plantillas de Código:

- **Prevención de Errores Comunes:** Android Studio ofrece asistentes y plantillas de código que ayudan a prevenir errores comunes. Al utilizar estas herramientas, se reducen los errores de sintaxis y se mejora la coherencia del código.

d) Pruebas Unitarias Integradas:

- **Detección Temprana de Problemas:** Android Studio facilita la escritura y ejecución de pruebas unitarias directamente desde el entorno de desarrollo. Esto permite detectar y corregir problemas antes de la implementación final, mejorando la confiabilidad y la robustez de la aplicación.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

4.1. Diseño de la interfaz de usuario.

En este apartado se detallara las funciones mas importantes de la aplicación mediante Mockups.

4.1.1. Navegación en la aplicación.

- Inicio de sesión en la aplicación: En primera instancia se mostrara la pantalla de inicio de la aplicación Calendario Académico USM. El usuario debe ingresar sus credenciales para iniciar sesión en la aplicación.



Figura 30: Inicio de sesión aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Registrar usuario en la aplicación: En caso de que el usuario no cuenta con una cuenta para iniciar sesión, tendrá la opción para crear una cuenta en la aplicación. El usuario debe ingresar su nombre completo, su correo electrónico, su RUT y su contraseña.



Figura 31: Registrar usuario aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Visualización del calendario: Al momento de iniciar sesión la aplicación redirigirá al usuario al calendario, mostrando el mes actual junto a los días, dentro de los días se encontrará el nombre de cada actividad.



Figura 32: Calendario aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Visualización de un día en específico: Al momento de hacer click en un día del calendario, se abrirá una nueva pantalla donde se mostrara información general como: El día junto con el mes, el tipo de actividad que se realiza, y la descripción que se llevara a cabo el mismo día.

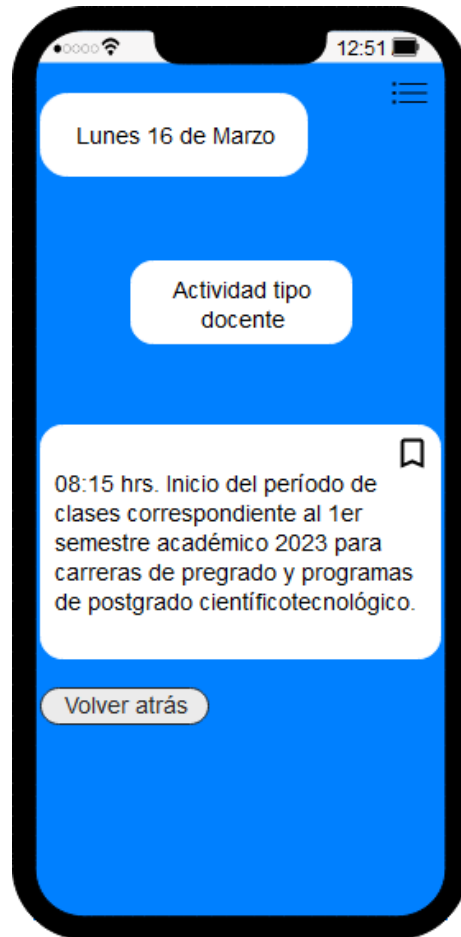


Figura 33: Calendario extendido aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Agregar actividad a favoritos: Al momento de seleccionar el icono de agregar a favoritos, la actividad se agregara a una lista de actividades favoritas que tenga el usuario.



Figura 34: Agregar actividad a favoritos aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Abrir menú: Al momento de seleccionar el icono de las tres barras, se desplegará un menú donde se encontrará una sección para acceder al perfil y otra para acceder a la lista de favoritos.

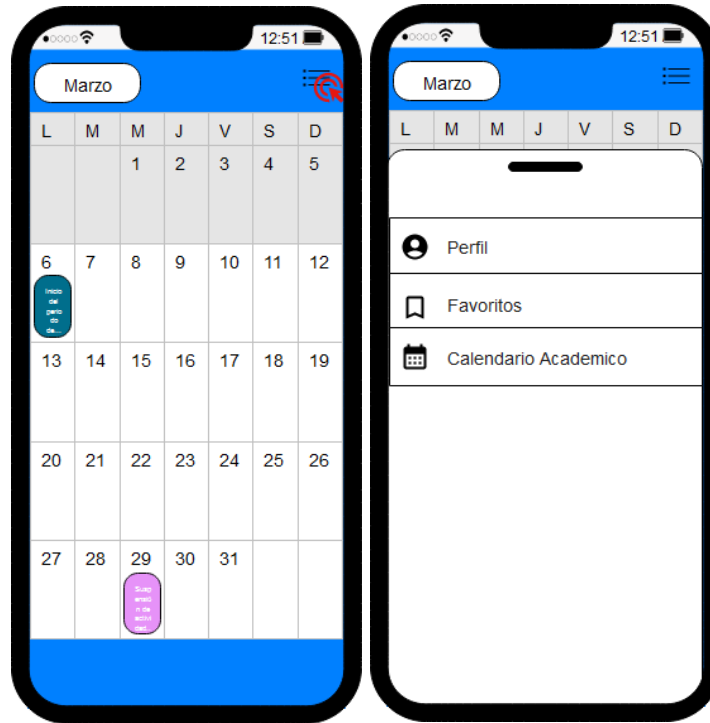


Figura 35: Desplegar menú aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Lista de favoritos: Al momento de seleccionar la opción de favoritos, se abrirá una nueva pantalla donde se listaran todas las actividades que están guardadas en favorito.

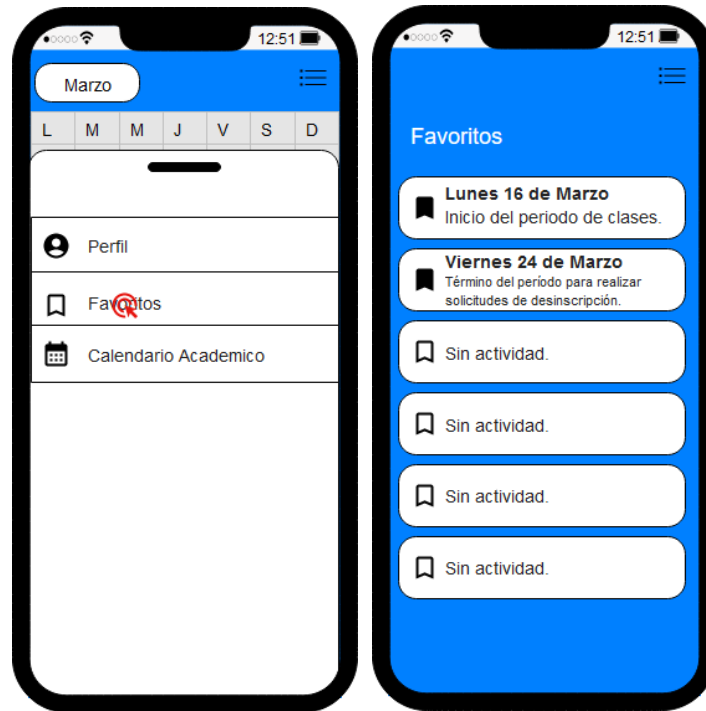


Figura 36: Ingresar a favoritos aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Perfil del usuario: Al momento de seleccionar la opción de perfil en el menú desplegable, se abrirá una nueva pantalla donde se encontrara toda la información del usuario junto a un icono en la esquina superior izquierda de configuración.

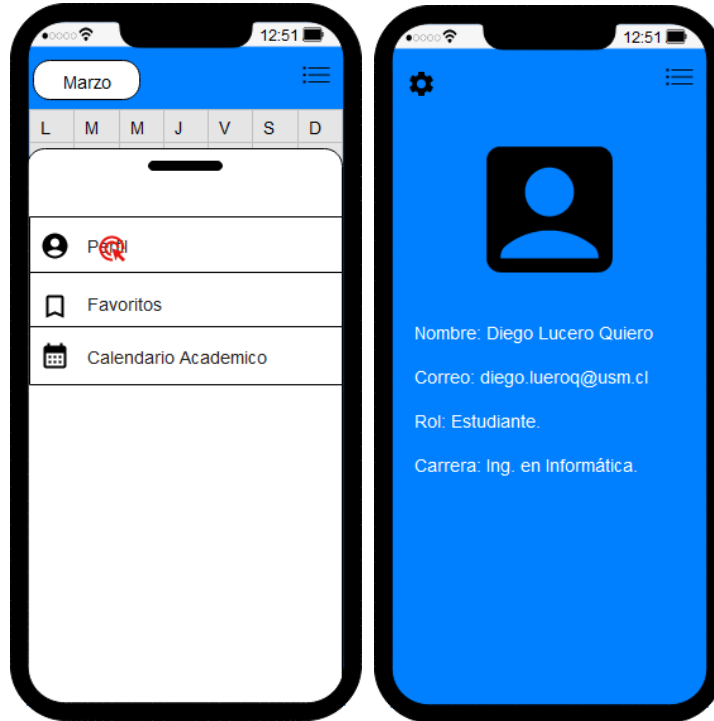


Figura 37: Ingresar a perfil aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- Configuración: Al momento de seleccionar el icono de configuración, se abrirá una pantalla nueva con opciones para cambiar contraseña, modificar el calendario en caso de que el usuario quiera cambiar colores, etc. Un switch para cambiar de modo oscuro o claro y un botón para volver atrás.

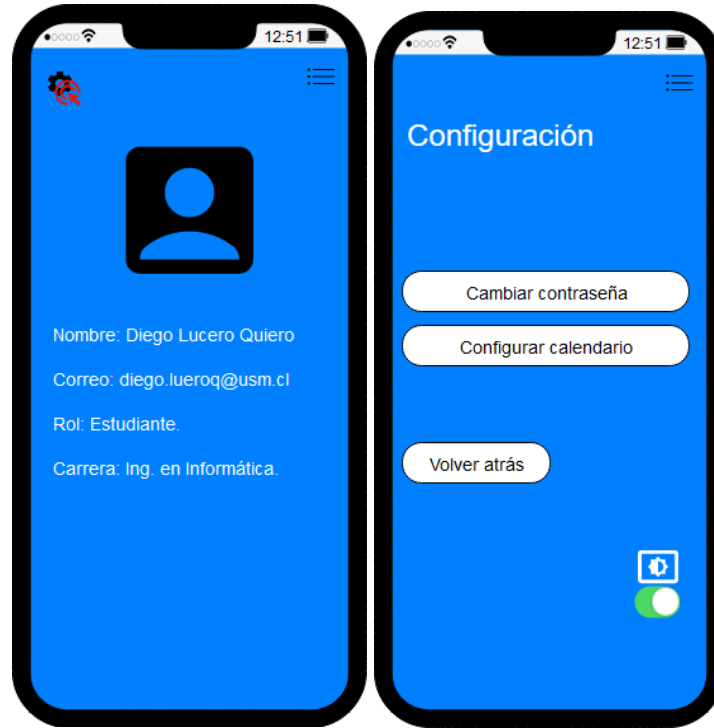


Figura 38: Ingresar a configuración aplicación Calendario Académico USM.
Fuente: Elaboración propia.

- **Buscar evento:** Se agrego una caja de búsqueda en la pantalla del calendario para que el usuario busque eventos a los que esta interesando mediante palabras claves (Por ejemplo: "Becas internas").



Figura 39: Búsqueda de eventos aplicación Calendario Académico USM.
Fuente: Elaboración propia.

4.2. Interacción con el calendario académico actual.

En esta sección se abordará el caso normal de un usuario de la Universidad haciendo uso del Calendario Académico en sus formatos actuales, ya sea en PDF o Excel.

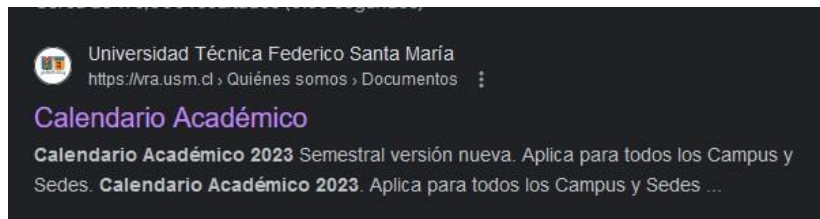
4.2.1. Acceso al calendario académico.

Flujo normal del uso del calendario por pasos.

- En primer lugar se debe ingresar al navegador de preferencia, en este caso utilizaremos Firefox.
- En segundo lugar se debe ingresar al buscador y hacer la siguiente búsqueda:



- Como tercer paso debemos realizar la búsqueda y seleccionar el primer link que aparece.

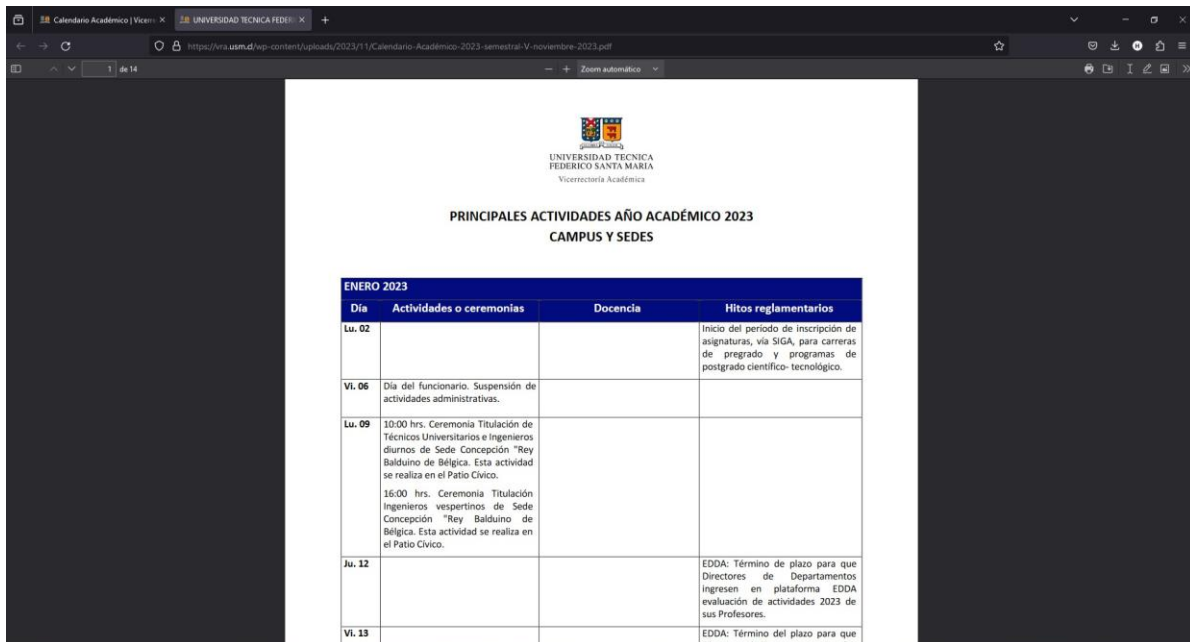


- Como cuarto paso, una vez ingresado a la pagina, nos encontraremos con lo siguiente:



Como se puede ver, la pagina oficial de la USM nos ofrece dos opciones para visualizar el calendario academico. En primera lugar tenemos el Calendario Academico del presente año semestral en su version nueva (Figura 2). En segundo lugar la pagina nos ofrece el Calendario Academico del año actual en su formato tradicional (Figura 1).

- Como quinto paso debemos elegir el formato que mas nos acomode, en este caso, se utilizara el Calendario Academico en su formato tradicional (PDF). Le damos click a la segunda opcion y comenzara la descarga del Calendario Academico (en otros casos como lo es ahora, el PDF se abre en una nueva ventana del navegador).



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
Vicerrectoría Académica

PRINCIPALES ACTIVIDADES AÑO ACADÉMICO 2023
CAMPUS Y SEDES

ENERO 2023			
Día	Actividades o ceremonias	Docencia	Hitos reglamentarios
Lu. 02			Inicio del periodo de inscripción de asignaturas, vía SIGA, para carreras de pregrado y programas de postgrado científico- tecnológico.
Vi. 06	Día del funcionario. Suspensión de actividades administrativas.		
Lu. 09	10:00 hrs. Ceremonia Titulación de Técnicos Universitarios e Ingenieros diurnos de Sede Concepción "Rey Balduino de Bélgica. Esta actividad se realiza en el Patio Cívico. 16:00 hrs. Ceremonia Titulación Ingenieros vespertinos de Sede Concepción "Rey Balduino de Bélgica. Esta actividad se realiza en el Patio Cívico.		
Ju. 12			EDDA: Término de plazo para que Directores de Departamentos ingresen en plataforma EDDA evaluación de actividades 2023 de sus Profesores.
Vi. 13			EDDA: Término del plazo para que Directores de Departamentos evalúen.

4.2.2. Comparación entre aplicación y método tradicional.

En este documento, se aborda de manera clara cómo una aplicación diseñada para reemplazar el Calendario Académico en su formato convencional puede mejorar significativamente la eficiencia de su uso en comparación con los métodos tradicionales.

A diferencia de los formatos actuales del Calendario Académico, la aplicación ofrece la conveniencia de descargarse de una sola vez, eliminando la necesidad de realizar descargas repetidas cada vez que se desee visualizar o consultar. Esto simplifica el acceso y evita la acumulación innecesaria de archivos en dispositivos como teléfonos móviles, computadoras y tabletas.

Otro aspecto importante a considerar es la capacidad de la aplicación para facilitar búsquedas específicas de eventos. A diferencia del calendario tradicional, que requiere explorar mes por mes y día por día, la aplicación permite una búsqueda más eficiente y menos tediosa, permitiendo a los usuarios encontrar rápidamente la información que necesitan.

Además de estas funciones destacadas en comparación con el calendario convencional, la aplicación ofrece características adicionales, como la capacidad de establecer recordatorios personalizados para actividades importantes, así como notificaciones automáticas para eventos significativos, como el inicio del período de clases o la suspensión de actividades académicas. Estas funcionalidades adicionales contribuyen a una gestión del tiempo más efectiva y a una mayor organización para los usuarios.

CAPÍTULO 5

CONCLUSIONES

En el transcurso del presente documento, se ha explorado a fondo la complejidad que posee manejar el calendario académico de la USM actualmente y se logró identificar una necesidad dominante de optimización. La entrega del calendario académico en sus formatos actuales como lo son en PDF y Excel, proveen información valiosa para los usuarios en general de la Universidad, a pesar de esto presenta un desafío significativo en términos de accesibilidad y practicidad para los usuarios.

La propuesta de una aplicación móvil dedicada a integrar el calendario académico y reemplazar el calendario en su formato actual emerge como una solución estratégica y centrada en el usuario. Esta alternativa busca superar las limitaciones actuales, ofreciendo una interfaz de usuario que simplifique la visualización de los eventos en general del calendario académico, además de proporcionar funcionalidades claves, como recordatorios personalizados, la búsqueda de eventos en concreto, alertas acerca de actividades importantes y el calendario en un formato más amigable.

Al implementar esta solución, se anticipa una mejora sustancial en la experiencia de los usuarios, facilitando el acceso a información relevante y contribuyendo a una gestión más eficiente del tiempo académico. Además, la aplicación móvil representa un paso hacia la modernización de los procesos universitarios, alineándose con las expectativas contemporáneas de accesibilidad y usabilidad.

La integración de una aplicación móvil para el calendario académico no solo aborda las limitaciones mencionadas, también anticipa beneficios tangibles para la comunidad universitaria en términos de eficiencia, comodidad y adaptabilidad a las demandas modernas. Este proyecto sienta las bases para una transformación significativa en la gestión de la información académica, promoviendo un entorno más accesible y conectado para estudiantes y funcionarios de la Universidad Santa María.

En la actualidad, la sociedad tiende a buscar soluciones accesibles y prácticas que estén al alcance de sus manos. En este contexto, la creación de una aplicación móvil para sustituir el calendario académico convencional adquiere una relevancia significativa. La idea es ofrecer a los usuarios la comodidad de acceder a la información que necesitan con tan solo un clic, evitando así el tedioso proceso de navegación y búsquedas complicadas.

Esta aplicación móvil se convierte en una herramienta intuitiva, donde los usuarios pueden encontrar fácilmente lo que buscan sin tener que dar vueltas innecesarias. La simplicidad de acceso a través de dispositivos móviles permite que la gestión de eventos académicos sea más eficiente, brindando a la comunidad una experiencia más fluida y directa. En resumen, se trata de poner la información al alcance inmediato de los usuarios, simplificando su experiencia y haciéndola más agradable.

5.1. **Discusión:**

En el transcurso del proyecto, la atención y dedicación se centraron de manera prioritaria en el diseño y desarrollo del modelo de datos, ignorando, por cuestiones de limitación de tiempo, el inicio del proceso de construcción de la aplicación móvil en si misma, Esta elección estratégica se baso en la premisa de que una base de datos solida y bien estructurada es fundamental para la eficacia y sostenibilidad a largo plazo de cualquier aplicación.

La decisión de priorizar el modelo de datos se justifica por la importancia de establecer una arquitectura robusta y escalable que pueda dar cabida a las complejidades inherentes al calendario académico. Un diseño cuidadoso de las tablas y las relaciones entre estas es esencial para garantizar una manipulación eficiente y precisa de la información académica en futuras etapas de desarrollo.

A pesar de enfocar este proyecto completamente en el modelo de datos, se reconoce la necesidad inminente de iniciar la fase de implementación de la aplicación. La construcción practica de la interfaz de usuario y la integración de las funcionalidad planificadas son pasos cruciales para poner a disposición de los usuarios finales una solución tangible y funcional.

Se espera que este enfoque secuencial, con una dedicación plena en el modelo de datos, genere beneficios significativos en la implementación posterior. El diseño sólido del modelo proporciona una base estructurada para la creación de la aplicación, reduciendo posibles complicaciones y mejorando la eficiencia en las etapas posteriores del desarrollo.

En el escenario en el que alguien decida utilizar este documento como punto de partida para iniciar el desarrollo de la aplicación, se le proporcionará una base sólida y completa que facilitará el comienzo del proceso. La meticulosa atención y enfoque integral dedicados a la elaboración del modelo de datos se traducen en una guía detallada, allanando el camino para la creación de la aplicación.

Este enfoque exhaustivo no solo simplificará el inicio del desarrollo, sino que también establecerá una estructura robusta para la aplicación. Al contar con un modelo de datos bien definido, el desarrollador tendrá una visión clara de las funcionalidades esenciales y la interconexión de los elementos clave de la aplicación. En última instancia, este enfoque integral sirve como cimiento sólido, allanando el camino para un desarrollo más eficiente y efectivo de la aplicación móvil.

5.2. **Resultados y validación.**

Se logro comprender detalladamente la gestión del calendario académico actual, destacando los desafíos asociados con los formatos de entrega y la falta de practicidad.

La propuesta sobre una aplicación móvil como solución se fundamenta en un diseño que

aborda específicamente las limitaciones actuales, ofreciendo una interfaz intuitiva y funcionalidades clave para cualquier tipo de usuario.

El modelo de datos detallado y bien estructurado se presenta como un resultado tangible, proporcionando una guía integral para la implementación de la aplicación.

Aunque no se ha iniciado la implementación práctica, la decisión estratégica de priorizar el modelo de datos proporciona una base robusta para la fase de desarrollo.

La propuesta de la aplicación móvil se destaca como un paso hacia la modernización, aliándose con las demandas contemporáneas de accesibilidad y usabilidad.

5.3. Contribución y aplicaciones del trabajo.

5.3.1. Identificación de Limitaciones Actuales:

- **Contribución:** La investigación identifica y analiza las limitaciones significativas en la gestión del calendario académico actual de la Universidad Santa María.
- **Aplicaciones Prácticas:** Comprender estas limitaciones proporciona la base para el desarrollo de soluciones específicas que aborden problemas reales enfrentados por la comunidad universitaria.

5.3.2. Propuesta de Solución Innovadora:

- **Contribución:** La propuesta de una aplicación móvil dedicada para integrar el calendario académico presenta una solución innovadora y centrada en el usuario.
- **Aplicaciones Prácticas:** La implementación de esta solución proporciona a los usuarios una herramienta accesible y práctica que simplifica la gestión de eventos académicos y mejora la experiencia general.

5.3.3. Priorización del Modelo de Datos:

- **Contribución:** La investigación prioriza el diseño detallado del modelo de datos como base para el desarrollo de la aplicación.

- **Aplicaciones Prácticas:** Un modelo de datos bien estructurado mejora la eficiencia en las etapas posteriores del desarrollo, reduciendo posibles complicaciones y asegurando una aplicación robusta y escalable.

5.3.4. Modernización de Procesos Universitarios:

- **Contribución:** La propuesta de una aplicación móvil no solo aborda problemas específicos sino que también busca modernizar los procesos universitarios.
- **Aplicaciones Prácticas:** La implementación exitosa contribuirá a una transformación más amplia, alineando la gestión de la información académica con las expectativas contemporáneas y mejorando la conectividad en la comunidad universitaria.

5.3.5. Guía Integral para Desarrolladores:

- **Contribución:** La investigación proporciona un modelo de datos detallado y bien estructurado, sirviendo como guía para aquellos que buscan iniciar el desarrollo de la aplicación.
- **Aplicaciones Prácticas:** Facilita el inicio del proceso de desarrollo al proporcionar una estructura sólida y una visión clara de las funcionalidades esenciales, mejorando la eficiencia y efectividad del desarrollo.

En conjunto, estas contribuciones no solo abordan desafíos específicos en la gestión del calendario académico, sino que también ofrecen soluciones prácticas y aplicables. La implementación exitosa de la aplicación móvil basada en estas contribuciones tiene el potencial de transformar la experiencia académica, mejorando la eficiencia y conectividad en la Universidad Santa María.

5.4. Impacto y recomendaciones.

El presente trabajo de investigación ofrece contribuciones significativas tanto a la organización como a los actores involucrados en la gestión del calendario académico de la Universidad Santa María.

1. Impacto en la organización:

- La propuesta de una aplicación móvil para integrar el calendario académico tiene el potencial de optimizar los procesos relacionados con la gestión de eventos académicos. La implementación exitosa de esta solución podría conducir a una mayor eficiencia en la difusión y acceso a la información aparte de reflejar un esfuerzo hacia la modernización de los procesos universitarios. Esto no solo mejora la experiencia del usuario, sino que también alinea la institución con las tendencias contemporáneas en tecnología educativa.

2. Aporte a los Actores Involucrados:

- Los estudiantes se benefician directamente al contar con una herramienta que facilita el acceso y la gestión del calendario académico. La aplicación proporciona información clara y oportuna sobre eventos cruciales, contribuyendo a una planificación académica más efectiva.
- El personal académico experimenta una simplificación en la comunicación de eventos y fechas relevantes. La aplicación móvil puede facilitar la coordinación de actividades académicas y mejorar la interacción con los estudiantes.

Recomendaciones para profundizar y futuras ampliaciones.

1. Investigación continua:

- Se recomienda realizar investigaciones adicionales para evaluar la aceptación y la eficacia de la aplicación móvil una vez implementada. Esto incluiría la recopilación de comentarios de los usuarios y la medición de indicadores clave de desempeño.

2. Ampliación de Funcionalidades:

- Para futuras ampliaciones, se sugiere considerar la incorporación de características adicionales, como recordatorios personalizados, integración con plataformas de aprendizaje en línea y la posibilidad de sincronización con otros calendarios digitales.

3. Escalabilidad y Adaptabilidad:

- Es crucial diseñar la aplicación de manera que sea escalable y adaptable a futuros cambios en el calendario académico o en las necesidades de los usuarios. Se debe considerar la posibilidad de actualizaciones periódicas para mantener la relevancia y la eficacia a lo largo del tiempo.

4. Colaboración con la Comunidad:

- Fomentar la colaboración con la comunidad universitaria durante todas las etapas de desarrollo. La retroalimentación continua de los usuarios puede ser invaluable para realizar ajustes y mejoras a medida que evoluciona la aplicación.

En resumen, el trabajo no solo presenta una solución innovadora para la gestión del calendario académico, sino que también establece una base sólida para futuras investigaciones y desarrollos. La aplicación propuesta tiene el potencial de generar un impacto positivo en la eficiencia y la experiencia académica en la Universidad Santa María.

ANEXOS

En los Anexos se incluye todo aquel material complementario que no es parte del contenido de los capítulos de la memoria, pero que permiten a un lector contar con un contenido adjunto relacionado con el tema.

REFERENCIAS BIBLIOGRÁFICAS

- [DiMarzio, 2016] DiMarzio, J. (2016). *Beginning android programming with android studio*. John Wiley & Sons.
- [Jemerov e Isakova, 2017] Jemerov, D. e Isakova, S. (2017). *Kotlin in action*. Simon and Schuster.
- [Lutkevich, 2022] Lutkevich, B. (2022). What is kotlin? Consultado el 8/10/2023.
- [Team, 2023] Team, K. (2023). Que es el modelo vista presentador? Consultado el 2/11/2023.