]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

user

[Dirección de correo electrónico]

Descripción breve

[Dibujar su lector con un resumen de la participación. Normalmente es un breve resumen del documento.   
Cuando esté listo para agregar contenido, haga clic aquí y empiece a escribir.]

[Título del documento]

[Subtítulo del documento]

**Introducción**

La era digital en la que vivimos ha generado numerosos cambios en el entorno educativo. Uno de los aspectos más destacados es la utilización de aplicaciones móviles para su desarrollo. Esta revolución tecnológica redefine sin lugar a dudas, la manera en que estudiantes, educadores y trabajadores de las mismas universidades interactúan, fomentando una mayor participación en su entorno. El uso de dispositivos móviles se ha vuelto cada vez más notorio, no solo debido a la digitalización escolar, sino también porque la sociedad ha evolucionado hasta el punto de hacer indispensables las herramientas tecnológicas en la vida cotidiana. Estos cambios han democratizado el acceso a la información y han facilitado diversas tareas académicas.

Una aplicación móvil, adaptada a cualquier institución, ofrece un acceso rápido y personalizado a información relevante, beneficiando a todas las partes interesadas en el proceso educativo. Permite un monitoreo constante de notificaciones, actividades, calificaciones y progreso individual, además de identificar áreas de mejora. De esta manera, el contenido se adapta para satisfacer las necesidades específicas de cada estudiante.

En Cuba, el desarrollo y la aceptación de aplicaciones móviles ha avanzado significativamente en la rama digital, llevando a cabo la creación de aplicaciones nacionales en áreas como la salud, el comercio y la educación. Esto ha posibilitado una mayor inclusión tecnológica y un acceso más eficiente a los servicios disponibles para los ciudadanos. La Universidad de Ciencias Informáticas (UCI), organización líder en la rama de la informática cuya principal misión es formar profesionales comprometidos con su país y altamente capacitados, así como desarrollar aplicaciones y servicios informáticos. Se basa en el modelo de formación estudio-trabajo y también apoya a la industria cubana de la informática. Desempeñando un papel fundamental en este avance, desarrollando múltiples soluciones tecnológicas destinadas a optimizar la gestión educativa y administrativa dentro de la institución. En dicha universidad, se realiza un proceso continuo de recolección de datos significativos para evaluar y caracterizar a los estudiantes universitarios. Esto tiene como objetivo formar juicios de valor y facilitar la comprensión de las situaciones relacionadas con el desarrollo de los educandos. Uno de los aspectos clave en el proceso de evaluación y caracterización integral es documentar la participación de los estudiantes en eventos estudiantiles.

En la UCI se vive una gran dinámica en cuanto a eventos estudiantiles, acogiendo varios como el Fórum de Historia, el Seminario Juvenil Martiano, los Juegos Mella, los Festivales de Artistas Aficionados, la Jornada del Ingeniero de las Ciencias Informáticas, y otros. La participación en estos eventos evidencia la formación integral de los estudiantes y contribuye a su evaluación y caracterización completa. Los eventos se dividen en diferentes categorías: científicos, culturales, deportivos y políticos, estos transitan por las fases de convocatoria, organización, realización, evaluación y anuncio de resultados.

La gestión y la participación en eventos son cruciales para el desarrollo académico y profesional tanto de estudiantes como de profesores en la universidad. Sin embargo, el proceso de gestión de los eventos estudiantiles en la Facultad de Tecnologías Libres (FTL) enfrenta varios desafíos: el gran volumen de información complica la recuperación de datos; las evidencias de la participación de los estudiantes provienen de diversas áreas y en distintos formatos (como Word, Excel o PDF[[1]](#footnote-1)), lo que impide la estandarización de la información. Además, la información sobre eventos pasados es escasa y difícil de acceder, ya que no existe un archivo histórico para consultas. Esto afecta la caracterización y evaluación integral de los estudiantes. Esta situación subraya la necesidad de una solución tecnológica que permita una gestión más eficiente y una mejor comunicación dentro de la comunidad universitaria. Esto trae consigo que la FTL necesite de una herramienta centralizada para la gestión de los eventos, específicamente los eventos científicos ya que posibilita a los estudiantes desarrollar habilidades, como trabajo en equipo, resolución de problemas y liderazgo, las cuales son fundamentales en el ámbito profesional. Estos eventos ofrecen la oportunidad de conocer las últimas tendencias y avances en el campo de la ingeniería, y permiten promover el pensamiento crítico y la capacidad de evaluar información de manera objetiva. Siendo de gran ayuda para obtener información importante dígase en el contexto del índice académico, cortes de integralidad de cada curso, y posibles vías para convalidar la tesis de grado. Al centrarse en estos eventos, el sistema de gestión se alinea con los objetivos principales de la formación de un ingeniero.

A partir de la **situación problemática** antes detallada, se puede definir el siguiente **problema de investigación:**

¿Cómo gestionar la información de los eventos científicos para contribuir al incremento de la accesibilidad, estandarización de la documentación y conservación de las evidencias en la Facultad de Tecnologías Libres a través de una aplicación móvil?

Acordando como **objeto de estudio:** los sistemas informáticos para la gestión de eventos científicos, estableciendo como **campo de acción:** la Facultad de Tecnologías Libres.

Para resolver el problema presentado, se propone como **objetivo general:** Desarrollar una aplicación móvil para gestionar la información asociada a los eventos científicos de manera que contribuya al aumento de la accesibilidad, estandarización de la documentación y conservación de las evidencias de los estudiantes en la Facultad de Tecnologías Libres.

A partir del objetivo general, se pueden precisar los siguientes **objetivos específicos:**

1. Realizar la fundamentación teórico-metodológica de la investigación permitiendo tener dominio sobre el tema a desarrollar.
2. Elaborar el análisis y diseño de la aplicación móvil para gestionar la información asociada a los eventos científicos en la FTL.
3. Desarrollar una aplicación móvil para gestionar la información asociada a los eventos científicos en la FTL.
4. Validar la solución propuesta a partir de la ejecución de pruebas a la aplicación móvil a desarrollar.

**Tareas de la investigación**

1. Descripción de los procesos de negocio asociados a los eventos científicos.
2. Realización de un estudio del estado del arte a nivel internacional y nacional, para conocer los referentes teóricos del objeto de estudio.
3. Selección de la metodología, herramientas de desarrollo, marcos de trabajo y lenguajes de programación a utilizar en el análisis, diseño e implementación de la aplicación móvil.
4. Definición de las funcionalidades y características principales que tendrá la aplicación móvil.
5. Implementación de las funcionalidades identificadas según las necesidades de los usuarios para obtener una aplicación móvil que cumpla con los requisitos determinados.
6. Validación de la aplicación móvil para gestionar la información asociada a los eventos científicos.

En el desarrollo de la investigación se emplearon los siguientes **Métodos de investigación científica:**

**Métodos del nivel teórico:**

* **Análisis-síntesis:** Consiste en la investigación de literatura y documentos existentes sobre aplicaciones móviles, gestión de eventos científicos, y estudios anteriores en contextos similares. Este método permitió identificar marcos teóricos relevantes. Valorar las tecnologías y metodologías más adecuadas para el desarrollo de la aplicación.
* **Modelación:** Se emplea para diseñar y verificar el sistema antes de su implementación real. A través de modelos conceptuales y simulaciones, se pudo analizar la arquitectura del sistema. Prever el comportamiento y rendimiento de la aplicación bajo diferentes escenarios.

**Métodos del nivel empírico:**

* **Entrevistas:** Dirigidas con expertos en gestión de eventos, desarrolladores de aplicaciones móviles y personal administrativo de la facultad. Las entrevistas en profundidad permitieron obtener una visión detallada y cualitativa de los requerimientos del sistema. Identificando desafíos y oportunidades en el desarrollo e implementación de la aplicación.
* **Observación:** Involucra la participación en eventos científicos para entender de primera mano cómo se gestionan. Este método brindo datos contextuales y realistas. Con una interpretación profunda del proceso actual de gestión y cómo puede mejorarse con la aplicación móvil.

La presente investigación se ha estructurado en tres capítulos, conformados de la siguiente forma:

**Capítulo 1: Fundamentación Teórica**

En este capítulo, se establecerán los conceptos básicos y el marco teórico que sustentan el desarrollo de aplicaciones móviles y la gestión de eventos científicos. Se aborda un estudio de sistemas homólogos; además se definen la metodología de desarrollo de software, los lenguajes y las herramientas para la modelación e implementación de la aplicación informática.

**Capítulo 2: Análisis y Diseño de la Propuesta de Solución**

Este capítulo se enfocará en la propuesta de solución basada en el análisis del problema previamente identificado. Se describirá en detalle el problema actual de la gestión de eventos científicos en la facultad, incluyendo entrevistas realizadas para obtener información sobre las dificultades que enfrentan estudiantes y profesores. Se definirán las funcionalidades que debe cumplir la aplicación y las características técnicas. Además, se presentará la arquitectura de la aplicación.

**Capítulo 3: Implementación y Pruebas de Validación**

Este capítulo acredita el desarrollo empírico de la aplicación, explicando cómo se llevó a cabo su implementación y cómo se probaron sus funcionalidades. Se puntualiza el proceso de codificación de la aplicación, incluyendo capturas de pantalla, descripciones de código clave y las herramientas utilizadas para el desarrollo. Se describirán las pruebas realizadas para asegurar que la aplicación cumple con los requisitos funcionales y no funcionales.

**CAPÍTULO 1: Fundamentación Teórica**

**Introducción**

Este capítulo establece el marco teórico que respalda el desarrollo de la aplicación móvil. Ofrece una revisión exhaustiva de conceptos clave. Además, se basa en estudios comparativos que demuestran cómo se utilizan los sistemas informáticos tanto a nivel nacional como internacional. Igualmente, se analizará la metodología de desarrollo de software[[2]](#footnote-2), los lenguajes y las herramientas para modelar e implementar la aplicación informática objeto del presente trabajo.

**1.1 Definición de conceptos**

Para entender mejor qué abarca esta investigación y precisar su objeto de estudio, a continuación, se presentan los conceptos fundamentales que sustentan su desarrollo.

**Aplicaciones Informáticas**

Son un tipo de software que permite al usuario realizar uno o más tipos de trabajo. Permiten la interacción entre usuario y computadora, dando opción a elegir opciones y ejecutar acciones. Existen innumerable cantidad de tipos de aplicaciones como los procesadores de texto y las hojas de cálculo, mientras que los sistemas operativos o los programas de utilidades que cumplen tareas de mantenimiento no forman parte de estos. Pueden haber sido desarrolladas a medida para satisfacer las necesidades específicas de un usuario o formar parte de un paquete integrado como el caso de Microsoft Office(Ramírez Carreño, 2022).

**Aplicaciones Móviles**

Las aplicaciones móviles o telefónicas, son programas diseñados específicamente para funcionar en dispositivos móviles como Smartphone[[3]](#footnote-3) y Tablets. Son herramientas que nos permiten realizar una amplia variedad de tareas, desde la comunicación con amigos y familiares hasta realizar transacciones bancarias o jugar videojuegos. Facilitan tanto tareas profesionales como de ocio, educativas y de acceso a servicios, mejorando la gestión de diversas actividades(Espinoza, 2022). Utilizan los recursos del dispositivo, como el procesador, la memoria y las conexiones a Internet, para ofrecer diversas funcionalidades.

**Sistemas operativos**

Un sistema operativo (SO) es un conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora. Es un software esencial que gestiona todos los recursos del hardware y proporciona una interfaz para que los usuarios puedan interactuar con el equipo, siendo una pieza fundamental de software sin la cual una computadora no podría funcionar correctamente(Villarreal, 2017).

**Sistema operativo Android**

Android es un sistema operativo móvil desarrollado por Google, basado en Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores. Su popularidad esta en rápido crecimiento, lo cual lo convierte en uno de los principales competidores en el mercado de dispositivos móviles(Judith & Salas, 2022). Fue desarrollado por Google convirtiéndolo en un sistema operativo móvil flexible, personalizable y con una gran comunidad de usuarios. Caracterizándose especialmente en ser un sistema operativo de código abierto, lo que significa que cualquier persona puede verlo, modificarlo y distribuirlo libremente; permitiendo que se convierta en el sistema operativo móvil más popular del mundo.

**API**

Una API (Interfaz de Programación de Aplicaciones) es como un intermediario que permite que diferentes programas o aplicaciones se comuniquen entre sí. Permite acceder a funcionalidades de otros servicios e integrar diferentes sistemas. En lugar de desarrollar todas las funcionalidades desde cero, puedes utilizar las API para aprovechar el código ya existente y probado de otros desarrolladores. Al mismo tiempo te ahorra tiempo y recursos, ayudando acelerar el desarrollo de tu aplicación y reducir los costos(Spositto, 2023).

**Sistema de gestión**

Un sistema de gestión es la herramienta que una organización utiliza para manejar las partes interconectadas de su negocio y alcanzar sus objetivos. Estos objetivos pueden abarcar una variedad de temas, como la calidad del producto o servicio, la eficiencia operativa, el rendimiento ambiental, la salud y la seguridad en el lugar de trabajo, entre otros(Alvarado, 2019).

**Información**

La información es un conjunto estructurado de datos procesados que forman un mensaje sobre un ente o fenómeno específico. Es conocimiento organizado que puede utilizarse para diversos fines. Además, la información otorga significado a las cosas y revela los patrones del pensamiento humano mediante códigos y conjuntos de datos(Thompson, 2008).

**Sistema de Gestión de Información**

Un Sistema de Gestión de Información se define como un conjunto de componentes interrelacionados que incluyen hardware y software que capturan, almacenan, procesan y distribuyen la información para apoyar la toma de decisiones, el control, análisis y visión en una institución. Un Sistema de Gestión de Información está formado por un conjunto de elementos integrados e interrelacionados que persiguen el objetivo de capturar, depurar, almacenar, recuperar, actualizar y tratar datos para proporcionar, distribuir y transmitir información en el lugar y momento en el que sea requerido en la organización(Hardy, 2019).

**Evento**

Un evento es básicamente algo importante que pasa en un contexto específico y que normalmente involucra a varias personas y actividades organizadas.

**Evento científico**

Un evento científico es un encuentro en el que investigadores, académicos y expertos de una determinada disciplina se reúnen para intercambiar sus descubrimientos, ideas y conocimientos(López-Giraldo, 2013).

Algunos ejemplos comunes de eventos científicos son:

**1. Conferencias:** Reuniones donde expertos presentan y discuten temas específicos.

**2. Congresos:** Encuentros formales que reúnen a profesionales para deliberar sobre temas de interés común.

**3. Seminarios:** Sesiones educativas donde se presentan y discuten investigaciones en desarrollo.

**4. Talleres:** Actividades prácticas donde los participantes aprenden y aplican nuevas habilidades.

**5. Simposios:** Reuniones para presentar y debatir investigaciones sobre un tema específico.

**1.2 Descripción de los procesos asociados a los eventos científicos**

En la UCI se llevan a cabo eventos científicos, organizados por el secretariado de la FEU[[4]](#footnote-4), la facultad o la universidad, según el nivel del evento.

En un evento científico, los estudiantes presentan ponencias de las cuales son autores. En la universidad, los eventos científicos incluyen la Jornada Científica Estudiantil, el Fórum de Historia y el Seminario Juvenil Martiano, entre otros. Las ponencias son agrupadas en comisiones, que cuentan con un tribunal encargado de evaluarlas. Este tribunal puede estar compuesto por estudiantes y profesores, con roles como secretario, vocal, presidente, miembro o alumno ayudante. Estos eventos pueden realizarse a nivel de facultad, universidad o nacional, permitiendo que los trabajos destacados se presenten en diferentes niveles del mismo evento. También se organizan competencias para medir habilidades y conocimientos. Entre estas competencias se incluyen: la Olimpiada de Idiomas, la Copa Pascal, la Copa de Bases de Datos y la Olimpiada de Matemáticas. Las participaciones pueden ser individuales o por equipos, y los resultados se basan en el conocimiento, la agilidad mental, la calidad de las respuestas y las habilidades demostradas por los estudiantes.

**1.3 Estudio del estado del arte**

El uso de aplicaciones móviles por parte de la población mundial evidencia la tendencia actual en la utilización de estos dispositivos, que se han convertido en imprescindibles para muchas personas. Diversos factores y circunstancias impulsan esta necesidad, con el objetivo de mejorar la calidad de vida y fomentar el desarrollo personal y en otros aspectos de los diferentes niveles poblacionales. Se integran perfectamente en nuestra vida cotidiana e influyen en la manera de pensar, de proyectarse, de interactuar con el mundo.

En el ámbito internacional educativo las aplicaciones móviles también desempeñan una función transcendental y para tener en cuenta resultados positivos en la enseñanza, aprendizaje y obtención de conocimientos se hace indispensable saber a quiénes van dirigidas. El público estudiantil exige condiciones y un estilo muy particular, requiere de aditamentos y de un tratamiento capaz de cubrir y cumplimentar sus objetivos básicos. Conocer las preferencias, hábitos y métodos de estudio y gestión de eventos sienta las bases para crear aplicaciones móviles que no solo atraigan a los usuarios, sino que también respondan a las necesidades de cada estudiante. Estas aplicaciones permiten gestionar sus estudios, calificaciones, eventos culturales, deportivos, científicos, entre otros. A continuación, se realiza un análisis de los sistemas homólogos encontrados que gestionan información sobre eventos estudiantiles de manera general.

**1.3.1 Sistemas de gestión a nivel internacional**

**Tu Fábrica de Eventos**

Tu Fábrica de Eventos es un software para eventos que ofrece una solución integral y personalizada para organizar cualquier tipo de evento, incluyendo los congresos científicos. Dentro de su amplia gama de módulos, recomienda el módulo de congresos científicos. El cual, permite gestionar todo lo relacionado con las contribuciones científicas. Es una aplicación informática que permite gestionar todos los aspectos relacionados con la organización de un evento científico, desde la fase de planificación hasta la fase de evaluación. Facilita el trabajo del comité organizador, del comité científico y de los asistentes al congreso, ofreciendo una plataforma online donde se puede acceder a toda la información y realizar todas las gestiones necesarias(TFE-web, 2021).

**Whova**

Whova es una plataforma integral diseñada específicamente para la gestión de eventos. Esta aplicación se ha posicionado como una solución completa para la organización de conferencias, congresos y otros eventos académicos de gran envergadura; con funciones como creación de agendas personalizadas, redes sociales dentro de la aplicación, encuestas y evaluación de eventos. En virtud de lo anterior dicho, cada participante puede construir su propia agenda según sus intereses, facilitando la navegación y la asistencia a las sesiones más relevantes. Esta plataforma fomenta la interacción entre los asistentes, la búsqueda de contactos, el envío de mensajes directos y la creación de grupos de discusión. Permite realizar encuestas y evaluaciones en tiempo real para medir la satisfacción de los participantes y recopilar feedback[[5]](#footnote-5) valioso para futuras ediciones del evento. También ofrece la posibilidad de transmitir en vivo las sesiones plenarias y conferencias, permitiendo la participación de aquellos que no pueden asistir presencialmente(Whova, 2018).

**CampusGroups**

Esta plataforma permite a las universidades crear una comunidad virtual para estudiantes, profesores y personal administrativo, donde les permite conectar, colaborar y organizar una amplia variedad de eventos, desde reuniones informales hasta grandes conferencias, y gestionar las inscripciones y la asistencia, entre otros. Al mismo tiempo esta aplicación ofrece herramientas de registro, gestión de asistentes, encuestas y comunicación. Fomenta la interacción y el intercambio de ideas entre estudiantes y profesores(CampusGroups,2020).

**Eventbrite**

Es una plataforma en línea para la creación y gestión de eventos. Ofrece una amplia variedad de herramientas y funcionalidades que facilitan la planificación, promoción y ejecución de eventos de todo tipo(Eventbrite,2018).

Es importante señalar que, aunque existen diversas soluciones informáticas para gestionar eventos científicos a nivel mundial, no es viable utilizarlas en la UCI. La integración con los sistemas de la UCI sería costosa, además no satisfacen las necesidades específicas de la universidad, como la relación y el soporte a los diferentes eventos.

**1.3.2 Sistemas de gestión a nivel nacional**

**Eventos UO**

Es la plataforma donde se organizan, gestionan y promueven los eventos científicos que tienen lugar en la Universidad de Oriente, Santiago de Cuba. Es un sistema concebido para adaptarse a las necesidades de los organizadores, y que propicia el intercambio de experiencias a los participantes. Ha sido desarrollado por el Grupo de Desarrollo de Software y Tecnología de la Universidad de Oriente (Universidad de Oriente, 2024).

**Multieventos**

Multieventos es un Sistema de gestión de eventos, encargada de gestionar los eventos científicos, específicamente los desarrollados en la UCI. Este sistema permite: gestionar usuario, encuestas, ponencias, convocatorias, talleres, roles y permisos. Además, el envío de notificaciones automáticas vía correos electrónico y hacer búsquedas generales(Coste, 2019).

**1.3.3 Resultado del estudio de las soluciones homólogas**

El estudio y la comparación de las herramientas mencionadas se resumen en la tabla a continuación con algunas de sus características fundamentales:

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | | **Software Libre** | **Aplicación Android** |
| Tu Fábrica de Eventos | | Sí | No |
| Whova | | Sí | Sí |
| CampusGroups | | Sí | Sí |
| Eventbrite | | Sí | No |
| Eventos UO | | No | No |
| Multieventos |  | No | No |

Tabla 1 Comparativa de sistemas Homólogos

**1.3.4 Conclusiones del estudio de las soluciones existentes**

Aunque se encontraron sistemas de gestión para eventos científicos, no se conoce ninguna herramienta de gestión de información para eventos científicos que solucione los problemas identificados en la Facultad de Tecnologías Libres. El estudio revela que estos solo resuelven parcialmente el problema de la investigación actual, ya que no cubren todo el proceso.

En base al análisis de las plataformas sólidas existentes que han establecido un modelo en la gestión de eventos, se concluye que estos sistemas no permiten la estandarización de la información. Para desarrollar una aplicación que contribuya a la gestión de la información de los eventos científicos, es fundamental identificar estas limitaciones. La información sobre eventos pasados es escasa y difícil de acceder debido a la falta de un archivo histórico, lo cual afectaría el proceso de la caracterización y evaluación integral de los estudiantes.

**1.4 Herramientas y tecnologías**

**1.4.1 Metodología de desarrollo de software**

Una metodología de desarrollo de software de alta calidad es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Estas metodologías guían a los equipos de desarrollo desde la concepción de una idea hasta la entrega final del producto.

Las metodologías de desarrollo de software proporcionan un ambiente estructurado para el desarrollo de software, ayudan a optimizar los recursos y reducir el tiempo de desarrollo. Garantizan que el software sea confiable, robusto y cumpla con los requisitos(Velásquez, 2019).

Entre las principales metodologías que existen, cada una se distingue por sus características y enfoques únicos. A continuación, se presentan las más populares:

**Metodología tradicional:** Las metodologías tradicionales de desarrollo de software, también conocidas como *metodologías en cascada*, se caracterizan por su enfoque lineal y secuencial. Cada fase del proyecto se completa por completo antes de continuar a la siguiente. Este enfoque se basa en una planificación detallada al comenzar el proyecto, asumiendo que todos los requisitos están claros y bien definidos desde el inicio, debido a que los cambios en los requisitos pueden ser costosos y difíciles de implementar(Velásquez, 2019).

**Modelo en Cascada (Waterfall)**

El modelo en cascada es el ejemplo más conocido de metodología tradicional. Se representa como una secuencia de fases, donde cada fase "cae" sobre la siguiente como una cascada(Velásquez, 2019).

**Metodologías Ágiles:** Las metodologías ágiles representan un cambio de modelo en la gestión de proyectos de desarrollo de software, en diferencia con los enfoques tradicionales como el modelo en cascada. En lugar de una planificación rígida y secuencial, las metodologías ágiles fomentan la flexibilidad, la colaboración y la adaptación a los cambios. Estas tienen un conjunto de prácticas, técnicas y herramientas que se centran en dividir el proyecto en ciclos cortos (sprints[[6]](#footnote-6)) de trabajo, lo que posibilita entregar valor al cliente de forma más rápida y sistemático. Promueven la colaboración continua entre los equipos de desarrollo y los clientes. Reconocen que los requisitos pueden cambiar durante el desarrollo y se adaptan a ellos de manera flexible. Poniendo en primer lugar siempre las necesidades del cliente, buscando entregar un producto que cumpla con sus expectativas(Lara & Figueroa, 2020).

Dentro de las principales metodologías ágiles que existen, se describen algunas de las más utilizadas:

* **Scrum:** Es una de las metodologías ágiles más usadas. Se basa en la formación de equipos auto organizados que trabajan en sprints de tiempo fijo(Bautista-Villegas, 2022).
* **eXtreme Programming (XP):** Un conjunto de prácticas ágiles que se centran en la calidad del código, la comunicación y la retroalimentación del cliente(Bautista-Villegas, 2022).

**Comparativa entre Metodologías Ágiles y Metodologías Tradicionales**

La **Tabla 2**, evidencia las principales diferencias entre las metodologías ágiles y las tradicionales. Estas diferencias que no solo afectan al proceso, sino también al contexto del equipo, así como su organización(Lara & Figueroa, 2020).

|  |  |  |
| --- | --- | --- |
| **Característica** | **Metodologías Ágiles** | **Metodologías Tradicionales** |
| **Enfoque** | Iterativo e incremental | Secuencial y lineal |
| **Planificación** | Flexible y adaptable | Detallada y rígida |
| **Requisitos** | Evolucionan a lo largo del proyecto | Se definen al inicio y son fijos |
| **Entrega** | Frecuente de productos funcionales | Entrega única al final del proyecto |
| **Roles** | Equipos auto organizados | Roles bien definidos |
| **Colaboración** | Alta, entre el equipo y el cliente | Baja, principalmente entre el equipo |
| **Documentación** | Justa, enfocada en el valor | Extensa y detallada |
| **Cambios** | Bienvenidos y gestionados | Dificultosos y costosos |
| **Valor al cliente** | Entrega temprana y continua de valor | Enfoque en cumplir con los requisitos iniciales |
| **Ejemplos de metodologías** | Scrum, Kanban, XP | Waterfall, RUP |

Tabla 2: Diferencias entre Metodologías Ágiles y Tradicionales.

**1.4.2 Selección de la metodología de software a utilizar**

La metodología eXtreme Programming (XP) es una herramienta poderosa para desarrollar software de alta calidad de manera ágil y adaptable. Aquí se presentan algunos argumentos sólidos a favor de la metodología XP:

* **Enfoque en el cliente:** XP introduce al cliente en el centro del desarrollo, garantizando que el software se ajuste a sus necesidades y expectativas. Esto se logra a través de una comunicación constante y una entrega frecuente de funcionalidades.
* **Alta calidad del software:** XP promueve la escritura de pruebas unitarias antes de escribir el código, lo que asegura que el software sea fuerte y confiable desde el principio. Además, la refactorización continua del código ayuda a mantenerlo limpio y sostenible.
* **Adaptabilidad a los cambios:** Al ser una metodología ágil, XP permite acomodarse rápidamente a los cambios de los requisitos del cliente. Esto es especialmente útil en entornos donde los requisitos pueden evolucionar velozmente.
* **Mayor productividad:** XP promueve la colaboración entre los miembros del equipo, ayudando a una mayor productividad y distribución del conocimiento.
* **Mayor satisfacción del equipo:** Al trabajar en un ambiente colaborativo y con un enfoque en la calidad, los integrantes del equipo acostumbran a estar más motivados y satisfechos con su trabajo.

La metodología XP es una estrategia de trabajo ágil que prioriza la calidad, la colaboración y la satisfacción del cliente. Es ideal para proyectos que requieren flexibilidad, adaptación a los cambios y entrega rápida de valor(Bautista-Villegas, 2022). Por todo lo anteriormente descrito, se decide emplear la metodología Programación Extrema (XP, eXtreme Programming) como guía para lograr el proceso de desarrollo de la propuesta de solución.

**1.4.3 Lenguaje de programación y marco de trabajo**

El mundo del desarrollo de aplicaciones móviles es muy dinámico y es capaz de ofrecer una amplia gama de lenguajes y herramientas para crear experiencias móviles excepcionales. Los lenguajes de programación más populares para el desarrollo móvil son Kotlin**,** Java y framework[[7]](#footnote-7) de tipo multiplataforma como lo son React Native y Flutter.

Para la implementación de la propuesta de solución se ha decidido utilizar como marco de trabajo **Flutter** dado que es un SDK (Software Development Kit[[8]](#footnote-8)) de código abierto, diseñado para desarrollar aplicaciones móviles de gran calidad para iOS y Android desde una única base de código. Esto significa que puedes construir una aplicación que se vea y se sienta nativa en ambas plataformas, ahorrando tiempo y recursos(Quisaguano Collaguazo, 2022).

Flutter es una tecnología híbrida desarrollada por Google y basada en el lenguaje Dart; permite crear aplicaciones nativas para Android y iOS en tiempo récord con una interfaz muy amigable basada en Material Design, con la que puedes hacer cambios en el código y ver el resultado en el emulador en tiempo real, mientras la app sigue en marcha(Quisaguano Collaguazo, 2022).

**Características fundamentales de Flutter:**

* **Rendimiento nativo:** Las aplicaciones de Flutter se compilan en código nativo, permitiendo asegurar un rendimiento fluido y rápido.
* **Widgets personalizables:** Flutter brinda una amplia gama de widgets prediseñados y personalizables, permitiendo crear interfaces de usuario únicas.
* **Hot Reload:** Una característica que permite ver los cambios en la aplicación en tiempo real, y agilizar el desarrollo.
* **Una sola base de código:** Escribe el código una vez y ejecútalo en múltiples plataformas.

Debido a las grandes potencialidades que posee, se ha decidido trabajar con Flutter. Esta plataforma favorece el desarrollo gracias a herramientas como Hot Reload[[9]](#footnote-9) y la sencillez de uso de sus widgets, permitiendo desarrollar aplicaciones más rápido. Además, permite crear interfaces de usuario fluidas y personalizadas, mejorando así la experiencia del usuario.

Por otro lado, se decide trabajar en el backend con el framework de desarrollo Django para gestionar la lógica y las interacciones de la aplicación, Python como lenguaje subyacente por su claridad y eficiencia, y Django Rest Framework (DRF) para crear y gestionar la API que facilita la comunicación entre el frontend y el backend. DRF permite que diferentes partes de una aplicación o diferentes aplicaciones se comuniquen entre sí. Ofrece características como vistas basadas en clases y vistas genéricas que simplifican las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y la autenticación y permisos para controlar el acceso a los datos.

**Dart: Un lenguaje diseñado para Flutter**

Flutter utiliza Dart como su lenguaje de programación principal. Dart es un lenguaje que sigue los comienzos de programación moderno orientado a objetos, y admite crear código bien estructurado y reutilizable, fácil de aprender para el desarrollo de interfaces de usuario y con un rendimiento sólido. Sus características lo hacen ideal para prevenir errores comunes en tiempo de compilación, su sintaxis es intuitiva y similar a otros lenguajes populares como Java y C#, simplificando la curva de aprendizaje. Dart compila el código a código nativo, lo que garantiza un alto rendimiento y una experiencia de usuario fluida(DartLangSpec,2020).

La combinación de Dart y Flutter ofrece una solución poderosa y eficiente para el desarrollo de aplicaciones móviles. Cuando se busca una plataforma para crear aplicaciones de alta calidad con un rápido desarrollo y una excelente experiencia de usuario, Flutter y Dart son una excelente opción.

**1.4.4 Herramientas de modelado**

Es un software que te da la oportunidad de crear diagramas y modelos visuales de tu aplicación. Estos modelos pueden representar desde la arquitectura general de la aplicación hasta la interfaz de usuario y los flujos de usuario(Castorena Peña, 2018).

**Las herramientas de modelado ofrecen:**

* **Visualización:** Permite ver la estructura completa de la aplicación de manera clara y precisa.
* **Comunicación:** Facilita la comunicación entre diseñadores, desarrolladores y otros miembros del equipo.
* **Planificación:** Ayuda a planificar y organizar el desarrollo de la aplicación.
* **Documentación:** Genera documentación de forma automática, agilizando el mantenimiento y la actualización de la aplicación.
* **Detección de errores:** Permite identificar posibles errores y problemas en el diseño antes de comenzar a codificar.

**Tipos de herramientas de modelado:**

* **Herramientas UML (Unified Modeling Language):** Estas herramientas utilizan diagramas UML para modelar sistemas orientados a objetos. Son ideales para modelar la arquitectura y el diseño de la aplicación.
* **Herramientas de Prototipado:** Permiten crear prototipos interactivos de la interfaz de usuario para obtener feedback de los usuarios.
* **Herramientas de mapeo de flujos de usuario:** Se utilizan para visualizar el flujo de la aplicación desde la perspectiva del usuario.

Para el desarrollo de la propuesta de solución se empleará Visual Paradigm for UML, una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: negocio, descripción de los requisitos, análisis y diseño orientados a objetos, implementación, pruebas y despliegue(Castorena Peña, 2018).

**Características Principales de Visual Paradigm for UML:**

* **Amplio soporte para diagramas UML:** Visual Paradigm cubre todos los tipos de diagramas UML, incluyendo diagramas de clases, casos de uso, secuencia, actividad, estado, componentes, despliegue entre otros.
* **Modelado de sistemas complejos:** Permite modelar sistemas de gran escala y complejidad, gracias a sus capacidades de gestión de grandes modelos y relaciones entre elementos.
* **Generación de código:** Puede generar código a partir de los diagramas UML, lo que agiliza el proceso de desarrollo de software.
* **Integración con otras herramientas:** Se integra con otras herramientas de desarrollo.
* **Personalización:** Permite personalizar la apariencia y el comportamiento de la herramienta para adaptarla a las exigencias específicas de cada usuario.

**Herramienta de Prototipado**

**Adobe XD** es una herramienta de diseño de experiencia de usuario (UX) y prototipado de interfaz de usuario (UI) desarrollada por Adobe Systems. Es ampliamente utilizada por diseñadores, desarrolladores y equipos de diseño para crear prototipos interactivos de aplicaciones móviles, sitios web y otras interfaces digitales(Wood, 2020).

La **Tabla 3**, evidencia las principales características y ventajas de utilizar Adobe XD.

|  |  |
| --- | --- |
| **Característica Principales** | **Ventajas** |
| **Diseño de Interfaz de Usuario** | Ofrece una amplia gama de herramientas para crear interfaces de alta calidad, incluyendo componentes, símbolos y capas. Facilita la creación de diseños profesionales y consistentes. |
| **Prototipado Interactivo** | Permite crear prototipos funcionales con enlaces y animaciones. |
| **Colaboración en Tiempo Real** | Posibilita el trabajo en equipo con funciones de colaboración en tiempo real. |
| **Generación de Código** | Puede exportar diseños a código. |
| **Facilidad de Uso** | Interfaz intuitiva y fácil de aprender. |
| **Versatilidad** | Se puede utilizar para diseñar una amplia variedad de interfaces. |
| |  |  | | --- | --- | | **Potentes Herramientas de Diseño** |  | | |  | | --- | | Ofrece una amplia gama de herramientas para crear diseños de alta calidad. | |

Tabla 3: Principales características y ventajas de utilizar Adobe XD.

Adobe XD es una herramienta poderosa y versátil para el diseño de experiencias digitales. Su facilidad de uso, integración con otras herramientas de Adobe y funciones de colaboración lo convierten en una opción popular para diseñadores y equipos de desarrollo.

**1.4.5 Sistema Gestor de Base de Datos**

Imagina una biblioteca gigante, pero en lugar de libros, guarda datos digitales. Esa biblioteca se conoce como **base de datos**. Una base de datos es una colección organizada de datos estructurados, diseñados para almacenar, gestionar y recuperar información de manera eficiente.

**SQL (Structured Query Language):** Es el lenguaje estándar para comunicarse con las bases de datos relacionales. Con SQL, se pueden crear, leer, actualizar y eliminar datos.

Este lenguaje estándar es tan popular debido a que es el lenguaje más utilizado para interactuar con bases de datos relacionales. Permite realizar consultas complejas y manipular grandes volúmenes de datos. La mayoría de los sistemas de gestión de bases de datos relacionales (SGBDR) como MySQL, PostgreSQL, lo soportan(Spector, 2018).

**SQLite**

SQLite es un sistema de gestión de bases de datos relacional (SGBD) que se destaca por su simplicidad y eficiencia. A diferencia de otros SGBD que requieren un servidor dedicado, SQLite se integra directamente en las aplicaciones, y significa que no necesitas una infraestructura adicional para utilizarlo. Esta característica lo convierte en una opción ideal para una amplia gama de aplicaciones, desde pequeñas herramientas hasta sistemas complejos(Sqlite,2020 ).

**Características Principales**

* SQLite almacena toda la información de la base de datos en un único archivo, lo que facilita su manejo y portabilidad. Esto significa que puedes mover fácilmente una base de datos SQLite de un dispositivo a otro sin necesidad de realizar configuraciones adicionales.
* Es un SGBD muy compacto y eficiente en cuanto a recursos, siendo ideal para dispositivos con limitaciones de hardware, como teléfonos móviles y tablets.
* Al igual que otros SGBD, SQLite organiza los datos en tablas relacionadas entre sí, y permite establecer vínculos entre diferentes tipos de información y realizar consultas complejas.
* Garantiza la integridad de los datos mediante transacciones. Si alguna parte de la transacción falla, todas las operaciones se deshacen, evitando así la corrupción de los datos.
* SQLite no requiere una configuración compleja. Ayudando a facilitar su uso, simplemente creas un archivo y comienzas a utilizar la base de datos.

SQLite es un SGBD ligero, eficiente y fácil de usar que se ha convertido en una opción popular para una amplia variedad de aplicaciones.

**1.4.6** **Entorno de desarrollo integrado**

Un Entorno de Desarrollo Integrado (IDE) es una aplicación que proporciona un conjunto completo de herramientas para escribir, limpiar y probar software. Es como un taller para programadores, donde tienen todas las herramientas necesarias a mano.

**Android Studio**

Es el entorno de desarrollo integrado (IDE) oficial proporcionado por Google para la creación de aplicaciones Android. Diseñado específicamente para este sistema operativo móvil, ofrece un conjunto completo de herramientas y funcionalidades que posibilitan el proceso de desarrollo(Bermúdez León, 2021).

**¿Por qué elegir Android Studio?**

* **Oficial y respaldado por Google:** Al ser el IDE oficial, Android Studio está constantemente actualizado con las últimas novedades y características de la plataforma Android.
* **Completo y versátil:** Ofrece todas las herramientas necesarias para el desarrollo completo de una aplicación, desde la creación de la interfaz de usuario hasta la publicación en la tienda.
* **Comunidad activa:** Cuenta con una gran comunidad de desarrolladores que comparten conocimientos, recursos y soluciones a problemas comunes.
* **Gratuito y de código abierto:** Android Studio es una herramienta gratuita y de código abierto, siendo accesible a cualquier desarrollador.

**Características Principales de Android Studio:**

* **Basado en IntelliJ IDEA:** Android Studio se construye sobre la sólida plataforma de IntelliJ IDEA, proporcionando una base estable y una gran cantidad de características avanzadas para la edición de código, refactorización, navegación y búsqueda.
* **Diseño visual de interfaces:** Android Studio incluye un editor visual que permite diseñar interfaces de usuario de manera sutil, arrastrando y soltando componentes. Esto agiliza el proceso de creación de diseños complejos.
* **Emulador integrado:** El emulador de Android incluido en Android Studio permite probar las aplicaciones en diferentes dispositivos virtuales con distintas configuraciones de hardware[[10]](#footnote-10) y software, sin necesidad de tener dispositivos físicos.
* **Depuración y perfiles:** Ofrece potentes herramientas de depuración y perfilado para identificar y solucionar problemas en el código de la aplicación, así como optimizar su rendimiento.
* **Integración con Gradle:** Utiliza Gradle[[11]](#footnote-11) como sistema de compilación, y proporciona una gran flexibilidad para configurar y personalizar los proyectos.
* **Soporte para múltiples lenguajes:** Además de Java, Android Studio admite otros lenguajes como Kotlin, C++.

Android Studio es una herramienta esencial para cualquier desarrollador que quiera crear aplicaciones Android de alta calidad. Su conjunto completo de características, su facilidad de uso y su integración con los servicios de Google lo convierten en la elección ideal para desarrollar aplicaciones móviles para el sistema operativo más popular del mundo(Bermúdez León, 2021).

**1.5 Bibliotecas para la gestión de estado de la aplicación**

**Retrofit**

Es una biblioteca que reduce la creación de solicitudes HTTP[[12]](#footnote-12) en aplicaciones Android, Java, Dart y Flutter. Define la estructura de las API de forma clara y precisa, generando automáticamente el código necesario para realizar las solicitudes. Esto hace que el código sea más limpio, reduce errores gracias a la tipificación fuerte y es sencillo de usar gracias a las anotaciones. Además, facilita la conversión entre objetos y datos(Android.Retrofit, 2018).

**Dio**

Es una biblioteca para Dart que favorece la realización de solicitudes HTTP. Es similar a Retrofit en su propósito principal, pero ofrece mayor flexibilidad. Dio permite realizar solicitudes HTTP de manera eficiente, interceptar y modificar tanto las solicitudes como las respuestas, y manejar errores de red. También, cuenta con interceptores personalizados para añadir lógica adicional y soporta múltiples formatos de datos. Gracias a su gran comunidad y documentación, Dio es una opción popular entre los desarrolladores de Flutter (Dio, 2018).

**Riverpod**

Es una solución para administrar el estado en aplicaciones Flutter. Riverpod gestiona el estado de forma reactiva, comparte datos entre diferentes partes de la aplicación y posibilita las pruebas. Esto se traduce en código más puro y organizado, mayor rendimiento y una gran flexibilidad (Riverpod, 2018).

**Conclusiones parciales del capítulo**

En conclusión, este capítulo ha establecido las bases teóricas y conceptuales para el desarrollo de una aplicación móvil innovadora que permita facilitar la gestión de información en eventos científicos en la Facultad de Tecnologías Libres. Se definieron las herramientas y tecnologías mayormente adecuadas para dar solución al problema planteado, teniendo en cuenta las características de estas y las necesidades descritas con anterioridad a las que da respuesta la propuesta de solución.

**CAPÍTULO 2:** **Análisis y Diseño de la Propuesta de Solución**

**Introducción**

Este capítulo detalla los procesos de negocio y la lista de reserva del producto. Se incluyen algunas historias de usuarios y tareas de ingeniería relacionadas. También se presentan los elementos para la estimación y planificación del primer plan de entregas. Además, se define la arquitectura del sistema, se muestran algunas tarjetas CRC[[13]](#footnote-13) y se describen los patrones de diseño y las personas implicadas en el sistema.

**2.1 Propuesta de solución**

Para abordar el problema presentado, se propone desarrollar una aplicación móvil que facilite la visualización detallada de los eventos programados mensualmente, incluyendo fechas, descripciones y requisitos de participación. Facilitar la inscripción de estudiantes en las convocatorias de los diferentes eventos, a través de un sistema de registro. Implementar un sistema de alertas que notifique a los usuarios sobre fechas límite de inscripción y eventos próximos. Proporcionar una herramienta de gestión de la información asociada a los eventos científicos que posibilita tener acceso a los resultados obtenidos por los estudiantes. Esta proposición ofrece una visión general de cómo desarrollar una aplicación móvil innovadora que supere las limitaciones de las plataformas existentes y ofrezca una experiencia más personalizada y enriquecedora para los estudiantes universitarios de la Facultad de Tecnologías Libres (FTL).

**2.2 Descripción de los procesos de negocio asociados a la organización de los eventos científicos.**

**Eventos científicos**

Los eventos científicos en la Universidad de Ciencias Informáticas (UCI) comienzan con la elaboración y lanzamiento de la convocatoria por parte de los dirigentes estudiantiles, a nivel de universidad o facultad. Los estudiantes interesados envían sus trabajos a la comisión organizadora, donde se agrupan por temáticas y se asignan a comisiones específicas. Los trabajos son evaluados el día del evento, y aquellos que cumplen con los requisitos son reconocidos como relevantes, destacados o con mención. Simultáneamente, en los otros eventos, los estudiantes compiten de manera individual o en equipo en diferentes categorías, buscando obtener los primeros lugares. Después de cada evento, la información generada es almacenada y divulgada, asegurando su accesibilidad para futuras referencias.

**2.3 Lista de reserva del producto**

En la lista de reserva del producto se establecen y priorizan las funcionalidades del sistema y se describen los requisitos no funcionales del software(Mendez, 2020). Estos requisitos detallan los servicios que ofrece el sistema y sus restricciones operativas, reflejando las necesidades de los clientes para resolver problemas como el control de dispositivos, la realización de pedidos o la búsqueda de información(Martín, 2019). A continuación, se presenta la lista de reserva del producto de esta investigación:

**2.3.1 Requisitos funcionales**

Tabla 1 Requisitos funcionales asociados al Módulo de administración y seguridad.

|  |  |  |  |
| --- | --- | --- | --- |
| **Número** | **Nombre** | **Descripción** | **Prioridad del cliente** |
| **RF1** | Autenticar usuario | Permite autenticar un usuario. | Alta |
| **RF2** | Registrar usuario | Permite registrar un usuario. | Alta |
| **RF3** | Modificar usuario | Permite modificar un usuario. | Alta |
| **RF4** | Eliminar usuario | Permite eliminar un usuario. | Baja |
| **RF5** | Listar usuarios por eventos | Permite listar usuarios. | Baja |

Tabla 2 Requisitos funcionales asociados al Módulo Inicial.

|  |  |  |  |
| --- | --- | --- | --- |
| **Número** | **Nombre** | **Descripción** | **Prioridad del cliente** |
| **RF6** | Crear eventos | Permite crear eventos. | Alta |
| **RF7** | Mostrar eventos | Permite mostrar eventos. | Alta |
| **RF8** | Modificar eventos | Permite modificar eventos. | Alta |
| **RF9** | Eliminar eventos | Permite eliminar eventos. | Alta |
| **RF10** | Listar próximos eventos | Permite listar próximos eventos. | Media |
| **RF11** | Listar eventos del mes | Permite listar eventos del mes. | Media |
| **RF12** | Buscar evento por su nombre | Permite buscar eventos. | Baja |
| **RF13** | Seleccionar eventos a participar | Permite seleccionar eventos a participar. | baja |
| **RF14** | Registrar tipo de resultado | Permite registrar tipo de resultado. | Media |
| **RF15** | Modificar tipo de resultado | Permite modificar tipo de resultado. | Media |
| **RF16** | Notificación de alerta | Permite notificar. | Baja |

Tabla 3 Requisitos funcionales asociados al Módulo de Eventos Científicos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Número** | **Nombre** | **Descripción** | **Prioridad del cliente** |
| **RF17** | Crear programa | Permite crear programa | Alta |
| **RF18** | Modificar programa | Permite modificar programa | Media |
| **RF19** | Mostrar programa | Permite mostrar programa | Alta |
| **RF20** | Eliminar programa | Permite eliminar programa | Media |
| **RF21** | Listar programa | Permite listar programa | Media |
| **RF22** | Crear comité organizador | Permite crear comité organizador | Alta |
| **RF23** | Listar comité organizador | Permite listar comité organizador | Alta |
| **RF24** | Modificar comité organizador | Permite modificar comité organizador | Media |
| **RF25** | Mostrar comité organizador | Permite mostrar comité organizador | Alta |
| **RF26** | Eliminar comité organizador | Permite eliminar comité organizador | Media |
| **RF27** | Registrar trabajo | Permite registrar trabajo | Alta |
| **RF28** | Mostrar trabajo | Permite mostrar trabajo | Alta |

**2.3.2 Requisitos no funcionales**

Los requisitos no funcionales (RNF) son características que hacen que el producto sea atractivo, usable, rápido o confiable. Son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras. Lograr la calidad de la especificación de RNF también evita que se introduzcan defectos que después de implementado el sistema son más costosos de resolver(Hernández, 2020). En resumen, los requisitos funcionales son esenciales en cualquier proyecto de desarrollo de software. Al definir claramente lo que el sistema debe hacer, se facilita el desarrollo y se garantiza que el resultado final satisfaga las necesidades del usuario.

Tabla 4 Requisitos no funcionales .

|  |  |  |
| --- | --- | --- |
| **Número** | **Tipo de RNF** | **Requisito no funcional** |
| **RNF1** | Hardware | Se debe disponer de un teléfono móvil o tableta. |
| **RNF2** | Usabilidad | La aplicación podrá ser usada por cualquier usuario que posea conocimientos básicos en el manejo de un dispositivo móvil. |
| **RNF3** | Usabilidad | La base de datos es interna |
| **RNF4** | Usabilidad | La aplicación se enfocará en registrar la información relacionada con el proceso de gestión de eventos científicos en la Facultad de Tecnologías Libres. |
| **RNF5** | Usabilidad | Solo se mostrará a los usuarios la información y las acciones que correspondan a sus responsabilidades o roles en el negocio, presentándolas en los menús pertinentes. |
| **RNF6** | Usabilidad | Las vistas del sistema deben mostrar en todo momento la acción que se está llevando a cabo, y los íconos deben estar representados por imágenes que correspondan a esa acción específica. |
| **RNF7** | Seguridad | El sistema de autenticación almacenará datos de los usuarios (usuario, contraseña, solapín). |
| **RNF8** | Seguridad | Se debe garantizar la seguridad mediante la codificación de usuarios y contraseñas para el acceso a la base de datos. |
| **RNF9** | Seguridad | El servidor de aplicaciones y de base de datos debe protegerse contra ataques para asegurar el código y la información. |
| **RNF10** | Desempeño | El sistema debe responder a las solicitudes de los usuarios en un promedio de menos de dos segundos, sin superar los cuatro segundos, excepto en la generación de reportes. |
| **RNF11** | Desempeño | El sistema debe manejar eficientemente la concurrencia de usuarios que interactúan con él. |
| **RNF12** | Interfaz | El sistema debe poder obtener información de sistemas externos para complementar su funcionamiento. |
| **RNF13** | Interfaz | El sistema debe proporcionar información sobre los eventos estudiantiles a sistemas externos a través de servicios web públicos. |

**2.4 Historias de usuario**

La historia de usuario es una técnica en XP para definir los requisitos del software. Consiste en tarjetas de papel donde el cliente describe brevemente las características necesarias del sistema, ya sean funcionales o no. El manejo de las historias de usuario es dinámico y flexible, permitiendo en cualquier momento reemplazarlas por otras más específicas o generales, añadir nuevas o modificarlas(Menzinsky,López, Palacio, Sobrino, Álvarez, Rivas 2022).

Las historias de usuario se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Describen, en una o dos frases, una funcionalidad de software desde el punto de vista del usuario, con el lenguaje que éste emplearía(Menzinsky,López, Palacio, Sobrino, Álvarez, Rivas 2022).

Tabla 5 Historia de usuario: Gestionar evento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Historia de Usuario** | | |  | |
| **Número: 1** | **Nombre de historia de Usuario:** Gestionar evento | | | |
| **Usuario:** Organizador, Administrador | |  | | |
| **Programador:** Yuniel Barroso González | | **Iteración asignada:** Primera | | |
| **Prioridad en el negocio:** Alta | | **Puntos estimados:** 0,5 | | |
| **Riesgo en desarrollo:** Bajo | | **Puntos reales:** 0,5 | | |
| **Descripción:** Esta actividad debe permitir crear, mostrar, modificar, listar y eliminar un evento. De un evento se registra: el nombre del evento, la fecha de inicio y de fin, el lugar donde será desarrollado, una breve descripción del evento, el comité organizador, la convocatoria y el cronograma. | | | |  |
| **Observación:** El usuario deberá proporcionar los archivos del cronograma y la convocatoria. | | | |  |

Tabla 6 Historia de Usuario: Registrar trabajo

|  |  |  |  |
| --- | --- | --- | --- |
| **Historia de Usuario** | |  | |
| **Número**: 2 | **Nombre de historia de Usuario:** Registrar trabajo | | |
| **Usuario:** Organizador, Administrador | |  | |
| **Programador:** Yuniel Barroso González | | **Iteración asignada:** Primera | |
| **Prioridad en el negocio:** Alta | | **Puntos estimados:** 0,5 | |
| **Riesgo en desarrollo:** Alto | | **Puntos reales:** 0,5 | |
| **Descripción:** Esta actividad debe permitir registrar un trabajo. Dado el conjunto de trabajos, se seleccionan algunos para ser registrados. | | |  |
| **Observación:** El estado del trabajo deberá cambiar a “Aceptado”. | | |  |

Tabla 7 Historia de Usuario: Modificar evento

|  |  |  |  |
| --- | --- | --- | --- |
| **Historia de Usuario** | |  | |
| **Número**: 3 | **Nombre de historia de Usuario:** Modificar evento | | |
| **Usuario:** Organizador, Administrador | |  | |
| **Programador:** Yuniel Barroso González | | **Iteración asignada:** Primera | |
| **Prioridad en el negocio:** Media | | **Puntos estimados:** 0,3 | |
| **Riesgo en desarrollo:** Medio | | **Puntos reales:** 0,3 | |
| **Descripción:** Esta actividad debe permitir modificar un evento en el cual se pueda cambiar algunos requisitos del evento, modificando detalles del evento. | | |  |
| **Observación:** El usuario es el encargado de realizar estas modificaciones. | | |  |

**2.5 Plan de entrega**

Se planificó una entrega correspondiente al módulo de eventos científicos. A continuación, se presenta el plan de entrega:

Tabla 8 Plan de entrega : Modulo de eventos Científicos

|  |  |  |
| --- | --- | --- |
| **Aplicación** | **Fin de la 1ra Iteración (3/9/2024)** | **Fin de la 2da Iteración (25/10/2024)** |
| Sistema para la gestión de los eventos científicos | Funcionalidades con prioridad alta. | Funcionalidades con prioridad media. |

**2.6 Estimación y planificación**

La estimación y planificación de iteraciones comienza con la lista de historias de usuario previamente definidas. Para cada iteración, se considera la prioridad establecida por el cliente. La duración asignada a cada historia de usuario se expresa en semanas, basado en su valoración real. Como resultado de estas historias de usuario recopiladas, se llegó a la siguiente planificación de la entrega:

Tabla 9 Estimación y planificación del plan de entrega.

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteraciones** | **Descripción** | **Estimación** | **Duración**  **total** |
|  | **Prioridad: Alta** |  |  |
|  | Autenticar usuario | 0,3 |  |
| Gestionar usuario | 0,5 |
| Gestionar eventos | 0,5 |
| Registrar trabajo | 0,5 |
| **1ra** | Seleccionar eventos a participar | 0,5 | 3.1  semanas |
| Gestionar comité organizador | 0,5 |
| Gestionar programa | 0,5 |
| Mostrar trabajo | 0,3 |
| **Prioridad: Media** | |  |  |
| **2da** | Registrar tipo de resultado | 0,3 | 1.2  semanas |
| Mostrar tipo de resultado | 0,3 |
| Gestionar funciones | 0,3 |
| Gestionar estado del trabajo | 0,3 |
| **Prioridad: Baja** | |  |  |
| **3ra** | Enviar notificación de alerta | 0,5 | 0,5 semanas |

**2.7 Tareas de ingeniería**

Resultan de la descomposición de las historias de usuario en unidades de trabajo adecuadas pare gestionar y seguir el avance de su ejecución(Menzinsky,López, Palacio, Sobrino, Álvarez, Rivas 2022). Estas asignaciones quedan descritas mediante las tareas de ingeniería. A continuación, se muestran algunas de las realizadas durante el desarrollo de la presente investigación:

Tabla 10 Tarea de Ingeniería No. 1.1 para HU "Gestionar evento".

|  |  |
| --- | --- |
| **Tarea de ingeniería** | |
| **Número tarea:** 1.1 | **Historia de Usuario:** No. 1 Gestionar evento |
| **Nombre tarea:** Diseñar las interfaces para gestionar un evento | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0,2 |
| **Fecha inicio:** 15/09/2024 | **Fecha fin:** 18/09/2024 |
| **Programador responsable:** Yuniel Barroso González | |
| **Descripción:** Desarrollar la interfaz de usuario, incluyendo los formularios necesarios para las opciones de crear, modificar, listar y eliminar un evento. | |

Tabla 11 Tarea de Ingeniería No. 1.2 para HU "Gestionar evento".

|  |  |
| --- | --- |
| **Tarea de ingeniería** | |
| **Número tarea:** 1.2 | **Historia de Usuario:** No. 1 Gestionar evento |
| **Nombre tarea:** Implementar las funcionalidades para gestionar un evento. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0,3 |
| **Fecha inicio:** 18/09/2024 | **Fecha fin:** 21/09/2024 |
| **Programador responsable:** Yuniel Barroso González | |
| **Descripción:** Implementar las funcionalidades de crear, modificar, listar y eliminar un evento. | |

Tabla 12 Tarea de Ingeniería No. 2.1 para HU “Registrar trabajo ”.

|  |  |
| --- | --- |
| **Tarea de ingeniería** | |
| **Número tarea:** 2.1 | **Historia de Usuario:** No. 2 Registrar trabajo |
| **Nombre tarea:** Diseñar la interfaz para registrar un trabajo. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0,2 |
| **Fecha inicio:** 21/09/2024 | **Fecha fin:** 23/09/2024 |
| **Programador responsable:** Yuniel Barroso González | |
| **Descripción:** Desarrollar la interfaz de usuario para registrar un trabajo, teniendo en cuenta la posibilidad de existir varios trabajos a la vez. | |

Tabla 13 Tarea de Ingeniería No. 2.2 para HU “Registrar trabajo”.

|  |  |
| --- | --- |
| **Tarea de ingeniería** | |
| **Número tarea:** 2.2 | **Historia de Usuario:** No. 2 Registrar trabajo |
| **Nombre tarea:** Implementar la funcionalidad registrar trabajo. | |
| **Tipo de tarea:** Desarrollo | **Puntos estimados:** 0,3 |
| **Fecha inicio:** 23/09/2024 | **Fecha fin:** 25/09/2024 |
| **Programador responsable:** Yuniel Barroso González | |
| **Descripción:** Implementar la funcionalidad de registrar trabajo, teniendo en cuenta la posibilidad de registrar varios trabajos a la vez. | |

**2.8 Arquitectura del Sistema**

La arquitectura del sistema es la estructura organizativa esencial de un software, que se refleja en sus componentes y sus interacciones. Abarca tanto los aspectos técnicos como los organizativos y define cómo se integran las diferentes partes del sistema para cumplir con sus objetivos funcionales y no funcionales(Cubero, 2020).

**2.8.1 Patrón arquitectónico**

Flutter según su documentación oficial posee una flexibilidad que brinda la libertad de elegir la arquitectura que mejor se adapte a las necesidades y preferencias. La arquitectura MVVM (Modelo-Vista-VistaModelo) es una excelente opción para el desarrollo de aplicaciones móviles con Flutter (ver figura 1), ya que es un patrón de diseño de software que se utiliza para separar la lógica de negocio de la interfaz de usuario, y además permite a los desarrolladores crear aplicaciones que son fáciles de mantener y escalar(Vega Ruíz, 2024)

En el contexto de Flutter, los modelos representan los datos de la aplicación. La vista se refiere a la interfaz que muestra los datos al usuario y recibe sus interacciones. El Vista-Modelo actúa como un puente entre los modelos y las vistas, gestionando la presentación y suministrando los datos necesarios para que la vista pueda mostrarlos*(*Vega Ruíz, 2024)*.*

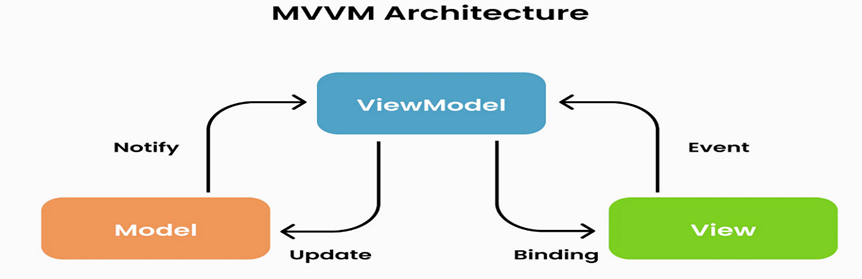
**

Figura 1. Arquitectura MVVM en Flutter

Modelo (Model): Representa los datos y la lógica de negocio de la aplicación. No tiene conocimiento de la interfaz de usuario ni de cómo se presentan los datos.

Vista (View): Es la interfaz de usuario que muestra los datos al usuario. La Vista se suscribe a eventos del VistaModelo para actualizarse cuando los datos cambian.

VistaModelo (ViewModel): Actúa como intermediario entre el Modelo y la Vista. Proporciona datos a la Vista y maneja la lógica de la interfaz de usuario. Utiliza comandos y enlaces de datos para comunicarse con la Vista y actualizarse automáticamente cuando los datos del Modelo cambian.

**2.8.2 Módulos del Sistema**

**Módulo de administración y seguridad**

Gestiona el funcionamiento y la seguridad del sistema, integrándose con los directorios activos. Se centra en definir qué usuarios y grupos pueden realizar ciertas operaciones en recursos específicos. Los roles del sistema se describen en el epígrafe 2.10. Este módulo también gestiona los usuarios y registra las trazas de sus acciones en el sistema.

**Módulo Común**

Administra las funcionalidades comunes a todos los tipos de eventos, como la creación y visualización del evento, los tipos de resultados, el comité organizador y el envío de notificaciones de alerta. Incluye una especialización: Módulo de eventos científicos.

**Módulo de eventos científicos**

Administra los eventos científicos, abarcando la gestión de los trabajos, el comité organizador y los resultados. Genera reportes nominales y estadísticos, como listados de participantes, del comité organizador y de eventos.

**2.9 Tarjetas CRC**

Las tarjetas CRC (clase­-responsabilidad-colaborador) se emplean para identificar y organizar las clases orientadas a objetos que son relevantes para el incremento actual de la aplicación móvil. El modelo CRC se compone de tarjetas índice CRC. Cada tarjeta CRC menciona el nombre de la clase, sus responsabilidades(operaciones) y sus colaboradores (otras clases a las que envían mensajes y de las que depende para lograr sus responsabilidades). Las colaboraciones implican una serie de relaciones (es decir, conexiones) entre clases del sistema(Lara & Figueroa, 2020). A continuación, se presentan ejemplos de tarjetas CRC realizadas.

Tabla 14 Tarjeta CRC Clase: Evento.

|  |  |  |
| --- | --- | --- |
|  | **Tarjeta CRC** | |
|  | **Clase:** Evento | |
| **Responsabilidades** |  | **Colaboraciones** |
| Crear Evento  Mostrar Eventos  Modificar Evento  Eliminar Evento  Registrar Trabajo |  | Comité Organizador  Resultado  Trabajo |

Tabla 15 Tarjeta CRC Clase: Programa.

|  |  |  |
| --- | --- | --- |
|  | **Tarjeta CRC** | |
|  | **Clase:** Programa | |
| **Responsabilidades** |  | **Colaboraciones** |
| Crear programa  Mostrar programa  Modificar programa  Eliminar programa  Registra tipo de resultado |  | Trabajo  Resultado  Evento |

Tabla 16 Tarjeta CRC Clase: Comité organizador.

|  |  |  |
| --- | --- | --- |
|  | **Tarjeta CRC** | |
|  | **Clase: Comité Organizador** | |
| **Responsabilidades** |  | **Colaboraciones** |
| Crear comité organizador  Mostrar comité organizador  Modificar comité organizador  Eliminar comité organizador | |  |

**2.10 Personas relacionadas con el sistema**

Una persona relacionada con el sistema es cualquier individuo que interactúa con él y obtiene resultados de uno o varios procesos. También se consideran aquellas personas que participan en los procesos sin obtener resultados de valor. Según su función en el negocio, se agrupan por roles, los cuales se describen en la siguiente tabla.

|  |  |
| --- | --- |
| **Personas relacionadas con el sistema** | **Descripción** |
| **Administrador** | Posee todos los permisos del sistema y es responsable de gestionar su seguridad y funcionamiento, incluyendo la administración de usuarios y trazas. También se encarga de gestionar los nomencladores del sistema. |
| **Organizador** | Tiene permisos para modificar eventos específicos en los que participa como miembro de la comisión organizadora. Es responsable de gestionar la organización del evento en el sistema. |
| **Usuario** | No tiene permisos para modificar; solo puede acceder a la información de los eventos registrados en el sistema. |

1. PDF\_ Del inglés Portable Document Format: formato de documento portátil. [↑](#footnote-ref-1)
2. Software: conjunto de instrucciones, datos o programas utilizados para operar computadoras y ejecutar tareas específicas. [↑](#footnote-ref-2)
3. Del inglés Smartphone: Teléfono inteligente. [↑](#footnote-ref-3)
4. FEU: Federación Estudiantil Universitaria. [↑](#footnote-ref-4)
5. Feedback «realimentación». Diccionario Español de Ingeniería (1.0 edición). Real Academia de Ingeniería de España. 2014. Consultado el 18 de mayo de 2014. [↑](#footnote-ref-5)
6. Sprints: un período corto y con un cuadro de tiempo en el que un equipo Scrum trabaja para completar una cantidad determinada de trabajo. [↑](#footnote-ref-6)
7. Framework: Se traduce como marco de trabajo [↑](#footnote-ref-7)
8. SDK (Software Development Kit): Conjunto de herramientas de desarrollo. [↑](#footnote-ref-8)
9. Hot Reload: Recarga caliente [↑](#footnote-ref-9)
10. Hardware: los dispositivos y equipos externos e internos que le permiten realizar funciones importantes a la computadora. [↑](#footnote-ref-10)
11. Gradle: herramienta de automatización para la creación de aplicaciones. [↑](#footnote-ref-11)
12. HTTP: Protocolo de transferencia de hipertexto. [↑](#footnote-ref-12)
13. Tarjetas CRC (Clase-Responsabilidad-Colaboración): son una herramienta utilizada en el diseño de software orientado a objetos. [↑](#footnote-ref-13)