

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

**ANÁLISIS E IMPLEMENTACIÓN DE SOLUCIÓN DE MENSAJERÍA Y ASUNTOS
ESCOLARES PARA EL COLEGIO LAICO DE VALDIVIA**

**Trabajo de titulación para
optar al Título Profesional
de Ingeniero de Ejecución
en Software.**

Alumno:

Carlos Alejandro Fica Muñoz

Profesor Guía:

Pedro Francisco Godoy Barrera

De todos los lujos concebibles, la muerte, bajo su forma inevitable e inexorable, es la más costosa. **G. Bataille**

RESUMEN

KEYWORDS: ESCOLAR, COMUNICACIÓN, COLEGIO, LIBRETA, COMUNICACIONES, ESCUELA, ASIGNATURAS, ADMINISTRACIÓN, MICROSERVICIOS, REACTJS, ELIXIR

El presente documento forma parte de un entregable para optar al Título Profesional de Ingeniero de Ejecución en Software. Su objetivo es aplicar técnicas de ingeniería en las distintas etapas de análisis y diseño de un sistema de software que brinde una solución informática a una problemática detectada, plausible de solventar con software.

Este trabajo presenta un análisis de ingeniería enfocado en la institución educacional privada Colegio Laico de Valdivia, identificando una problemática crítica en la distribución y gestión de las comunicaciones entre sus funcionarios y los apoderados. Este desafío impacta negativamente en la fluidez y efectividad de los procesos informativos esenciales para garantizar el adecuado seguimiento y apoyo al desarrollo educativo de los niños y jóvenes matriculados en la institución.

El presente informe se estructura en los siguientes capítulos:

Capítulo 1: Aspectos relevantes del proyecto y su gestión. Se describe brevemente la estructura social y operativa del Colegio Laico de Valdivia, así como el trabajo de ingeniería realizado para recopilar los procesos operativos susceptibles de ser implementados en una solución informática.

Capítulo 2: Aspectos relevantes del análisis para la alternativa seleccionada. Se detallan los requerimientos del proyecto y se definen los casos de uso derivados de ellos.

Capítulo 3: Diseño arquitectónico y prototipado. Se presenta el diseño arquitectónico del sistema, ilustrando la composición de componentes, el manejo de datos y las interacciones de los casos de uso más importantes. Además, se incluyen prototipos para representar visualmente la solución de software terminada.

INTRODUCCIÓN

El Colegio Laico de Valdivia, es un colegio particular subvencionado enclavado en la ciudad de Valdivia. Fue fundado en marzo de 2004 con la premisa de otorgar una opción educativa diferente enfocado en una visión de tolerancia, respeto por la diversidad con pilares arraigados en la enseñanza laica. Desde el inicio de su funcionamiento el Colegio Laico en adelante *El Colegio*, ha tenido una gran receptividad y comunicación con los padres y apoderados de la comunidad educativa, esto principalmente por la cercanía generada, en parte por el tamaño del colegio y además de la larga estancia promedio de las familias en *El Colegio*.

En el último tiempo especialmente en el periodo postpandemia, se incrementó el uso de distintos medios de comunicación digitales, tales como el Ms-Teams, mail de padres/apoderados y en forma más masiva WhatsApp; el cual permite una comunicación instantánea cuando esta es posible. Lo anterior ha dejado en detrimento el uso de la libreta de comunicaciones en papel, que por reglamento [1] es el medio oficial para las comunicaciones entre el colegio y el hogar; en parte por las ventajas que ofrecen las comunicaciones digitales por sobre el papel (masividad, instantaneidad, consistencia)

El presente documento analiza las necesidades que pueden ser solventadas mediante software en *El Colegio* y confluir en una "*Solución de mensajería y asuntos escolares para el Colegio Laico de Valdivia*" que da respuesta a las necesidades de comunicación entre los distintos entes que conformantes de la comunidad educativa de *El Colegio*.

Detallando lo mencionado anteriormente, esta solución de software contempla trazar como objetivo general, unificar las vías de comunicación escolar en un software integral y referente a los objetivos menores de este proyecto son: que esta solución de software sea el canal oficial para las comunicaciones, prescindir gradualmente del uso de la libreta de papel y por último disponer de los indicadores escolares disponibles para consulta de forma instantánea.

Para acometer este proyecto, se contempla el uso de la metodología cascada, que tiene la ventaja de un enfoque sistemático y secuencial [2], además de ser apta, para los requerimientos definidos y estables del presente proyecto.

CAPÍTULO 1: ASPECTOS RELEVANTES DEL PROYECTO Y SU GESTIÓN

1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN

1.1.1. Antecedentes de la Empresa u Organización.

1.1.1.1. Antecedentes Generales.

El Colegio Laico de Valdivia, de en adelante CL, es una organización educativa privada subvencionada, inscrita con el ROL RBD22459 [3] en el Ministerio de educación de Chile y el ente controlador administrativo y financiero es la *Corporación Educacional Colegio Laico de Valdivia*. CL está ubicado en la calle González Bustamante 1991, Ciudad de Valdivia, Región de Los Ríos. CL se fundó el primero de marzo de 2004 con una matrícula no superior a los 120 alumnos [4].

El colegio es particular subvencionado operando bajo el amparo de la ley 20.845 [5], por lo cual es sin fines de lucro. Cuenta con 14 niveles de educación que incluyen preescolar (prekínder - kínder) básica y media. Actualmente cuenta con 465 alumnos matriculados con un promedio de 33 alumnos por curso; cuenta con 31 docentes y 15 funcionarios de la educación.

La estructura organizacional está encabezada por la Entidad Sostenedora, responsable última de la institución. La Dirección gestiona la institución, y bajo ella se encuentran dos ramas principales: la Unidad Técnico-Pedagógica, centrada en lo académico, y los Funcionarios Administrativos, encargados de la gestión interna. La Unidad Técnico-Pedagógica se subdivide en Docentes, Funcionarios Profesionales y Funcionarios Paradocentes, cada uno con roles específicos en el apoyo a la educación.

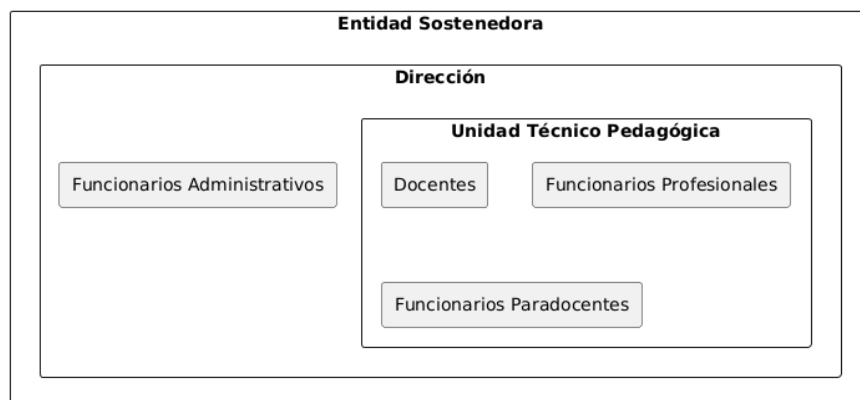


Ilustración 1-1 Diagrama de Jerarquía
Fuente: Elaboración Propia

1.1.1.2. Proyecto Educativo

Se desprende de forma textual: *"El Colegio Laico Valdivia, es un Proyecto Educativo cuya orientación académica se orienta hacia la formación integral bajo los principios Humanistas Laicos que propone la Francmasonería [6], congruentes ellos, con los Objetivos Transversales que plantea la Reforma Educacional. Se busca, entonces, que el ser humano tenga la posibilidad de emprender con libertad de conciencia la construcción de su vida..."* [4]

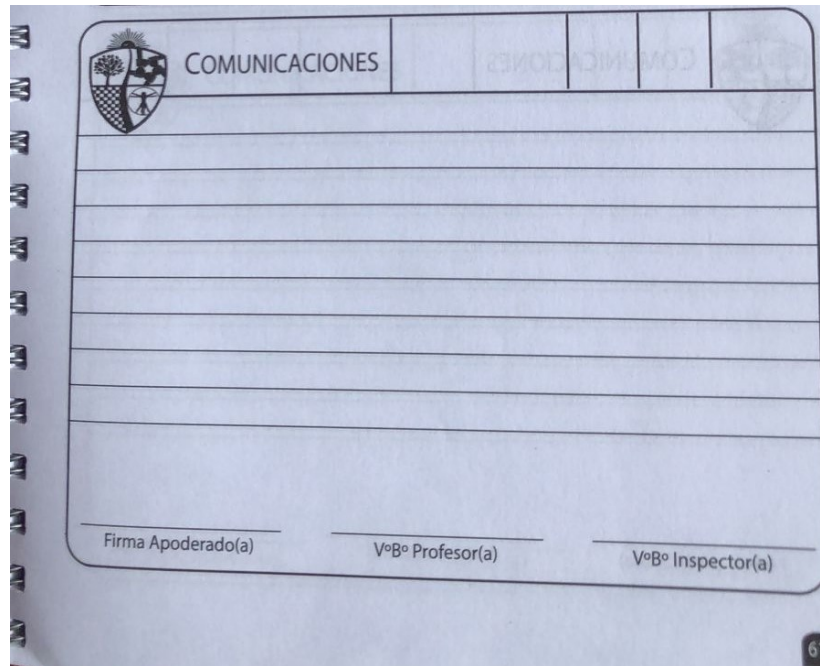
1.1.1.3. Misión

Se desprende de forma textual: *"La misión de su Proyecto educativo se centra en dos ejes esenciales los cuales comprenden: el desarrollo integral del alumno y un trabajo basado en la afectividad. Bajo las anteriores premisas el colegio se ha destacado como una institución educativa comprometida con la formación de sus estudiantes..."* [4]

1.2. DESCRIPCIÓN DE LA SITUACIÓN SIN PROYECTO

Desde su creación, CL mantiene una comunicación amena con los padres y apoderados [4]. Según el análisis mediante entrevistas, se describen las siguientes modalidades de comunicaciones al hogar:

- **Presencial**, frecuente en los traslados de los alumnos a clases, reuniones de apoderados, cuando el padre o apoderado asiste al establecimiento.
- **Telefónica**, para comunicados con materias urgentes y/o prioritarias.
- **Mail**, utilizada informar acerca de eventos sociales del colegio y gestiones importantes, que involucran a toda la comunidad educativa
- **Libreta de comunicaciones**, método usado generalmente para citas a reuniones, recordatorios de tareas, avisos de atrasos y justificaciones de inasistencia. Esta libreta tipo agenda, contiene un apartado diseñado (Imagen 1-1) para efectuar las comunicaciones desde El colegio al hogar, donde docentes y funcionarios dejan mensajes para los padres y apoderados. Se incluyen en esta libreta los siguientes elementos: registro de calificaciones, horario de clases y agenda mensual. Estos ítems deben ser completados por cada alumno durante el inicio del año escolar.



Formulario de comunicaciones del Colegio Laico. El formulario tiene un encabezado con el logo del colegio a la izquierda y el título "COMUNICACIONES" a la derecha. Debajo del encabezado hay un espacio con líneas horizontales para escribir. En la parte inferior del formulario, hay tres líneas para firmas, etiquetadas como "Firma Apoderado(a)", "VºBº Profesor(a)" y "VºBº Inspector(a)". En la esquina inferior derecha del formulario, hay un pequeño círculo con el número "61".

Imagen 1-1 Módulo de comunicaciones - Colegio Laico
Fuente: Colegio Laico

Como se indica en el reglamento de convivencia [1] del colegio Laico, las comunicaciones están normadas por el siguiente orden de prioridad: teléfono del apoderado, a través de la agenda escolar, correo electrónico institucional. En primera instancia y según la información del reglamento de convivencia, las comunicaciones con padres y apoderados tienen las siguientes finalidades:

- Entrevista de información general
- Entrevista de acompañamiento pedagógico al alumno
- Entrevista para informar medidas de resguardo
- Entrevistas con Psicóloga
- Entrevistas de seguimiento

Desde la implementación de las medidas COVID en los colegios y la paulatina “nueva normalidad” que se tradujo en el retorno a clases bajo restricciones, muchos medios de comunicación entre el colegio y el hogar afloraron en detrimento de los canales mencionados anteriormente, esto debido en parte a la contingencia producida por las medidas COVID implementadas por el Ministerio de Salud. Como resultado, canales de comunicación de mensajería instantánea como WhatsApp y Telegram comenzaron a ser usados de forma extraoficial en el periodo de pandemia. Con estas herramientas Padres y apoderados junto con profesores intercambiaban mensajes informativos sobre los diversos quehaceres de los alumnos.

Pasado el tiempo de pandemia COVID y posterior levantamiento de las restricciones de aforo en los recintos educativos, se mantuvieron los canales de comunicación extraoficiales vía mensajería instantánea (WhatsApp – Telegram) de padres y profesores, en específico para asuntos relacionados con actividades delimitadas de cada curso y asuntos inmediatos tales como: eventos de cada curso, compra de materiales escolares, eventos a beneficio, entre otros.

En el caso para obtener la información acerca de los registros académicos ya sean notas, anotaciones de conducta, estas son entregadas de forma trimestral mediante correo electrónico.

En resumen, se evidencia que hay distintos motivos de comunicaciones, los cuales no están alineados y normados, dando lugar a una fragmentación e informalidad propias por el uso de distintos canales de comunicación.

1.3. PROBLEMAS DETECTADOS.

A continuación, se enumerarán las principales deficiencias apreciadas en el análisis realizado a los distintos tipos y canales de comunicaciones en El colegio:

Fragmentación de las comunicaciones:

Las comunicaciones generadas desde El colegio al hogar, dependiendo de su naturaleza ya sean: informativas, informativas pedagógicas, psicológicas, generales (reuniones de apoderados, eventos, colectas) Se usan distintos canales de comunicación para su cometido, desde el canal primario oficial, que es la llamada telefónica cuando es un asunto particular de algún alumno; hasta el canal secundario de correo electrónico en caso de asuntos generales. También cabe mencionar que se realizan comunicados en los distintos "grupos de WhatsApp" de los distintos cursos para difundir estas comunicaciones generales. Resumiendo lo anterior, se evidencia que El colegio no tiene un canal oficial de comunicación para las diferentes necesidades, originando fragmentación y poco orden a los comunicados emanados desde El colegio.

Informalidad de las comunicaciones:

La adopción de los sistemas de mensajería instantánea en periodo de pandemia COVID por parte de la comunidad escolar, creo una dependencia de uso, gracias a la facilidad de configuración y masividad de sus usuarios. Cualquier apoderado o funcionario del colegio puede fácilmente entablar comunicaciones en grupos de WhatsApp o Telegram y así transmitir distintos comunicados y/o información referente a cualquier ámbito atingente a lo escolar. Lamentablemente la creación de estos "grupos" de chat solo esta normada por los miembros que conforman el chat grupal y no cuentan con el visado de El colegio para su funcionamiento, lo cual resta formalidad en los comunicados. Se evidencia frecuentemente que las temáticas de los estos "grupos" de chat no mantengan integridad escolar académica, ya que no cuentan con una moderación transparente para el Colegio, resultando en informalidad de las comunicaciones que puedan resultar en tales "grupos" de chat.

Libreta de comunicación sin uso:

El uso de la libreta de comunicación u agenda escolar ha quedado en segundo plano, respecto de otros canales de comunicación, tales como la llamada escolar, mensajería instantánea u el correo electrónico institucional. Esto en parte a la adopción de las nuevas tecnologías, sin contar además que la escritura de un comunicado en la libreta de comunicaciones es algo en desuso y puede ser algo complejo de implementar en los cursos compuestos por alumnos que no dominan la lectoescritura.

Poca información de indicadores escolares para apoderados:

Los indicadores escolares tales como: notas, asistencia, anotaciones de conducta, entre otros son comunicados / entregados a los apoderados bajo los siguientes esquemas:

- Entrega en las reuniones de apoderados trimestrales. En este caso son entregadas vía papel y además publicadas en el buzón de MSTeams personal de cada alumno
- Solicitud particular de indicadores por parte del apoderado. En este caso es el padre o apoderado del alumno que realiza la solicitud de indicadores a una autoridad competente (docente, inspector, secretaria) el cual el funcionario realiza la gestión y posterior entrega de los indicadores al apoderado solicitante.

En ambos casos los indicadores ya sea de conducta o notas, son comunicados con una periodicidad amplia, en donde los padres y apoderados no pueden tener con una retroalimentación de estos indicadores de forma rápida.

1.4. OBJETIVOS

1.4.1. Objetivo General

El objetivo general de este trabajo es unificar y mejorar las comunicaciones escolares entre el Colegio y el hogar, mediante una solución de mensajería instantánea, para su uso entre los entes educativos del Colegio: docentes, funcionarios y padres y/o apoderados del colegio con el fin ulterior de disponer información académica de forma instantánea para la comunidad educativa.

1.4.2. Objetivos Específicos

Teniendo en cuenta que el objetivo general del presente proyecto es unificar las comunicaciones escolares, mediante una solución de software, se desprenden los siguientes objetivos específicos:

- a) Desarrollar una solución de software robusta y confiable, para que sea considerada como el canal oficial de comunicaciones desde el colegio al hogar.
- b) En base al punto anterior, ofrecer una alternativa para prescindir de la libreta de comunicaciones de papel usada actualmente en el colegio.
- c) Desarrollar una plataforma de información de indicadores escolares de los alumnos, ya sean notas y anotaciones conductuales, de forma instantánea a los apoderados, reduciendo el tiempo involucrado actualmente en el proceso de entrega de los indicadores escolares.

1.5. REQUERIMIENTOS Y REQUISITOS DEL USUARIO.

Posterior al análisis realizado en conjunto con algunos apoderados y funcionarios de El colegio, quedó en claro que gran parte de las funcionalidades requeridas residen en la facilidad de la mensajería instantánea que está presente en las aplicaciones WhatsApp y Telegram, por lo cual gran parte de los requerimientos y requisitos próximos a presentar se inspiran en las aplicaciones mencionadas anteriormente.

A continuación, se presentará un compendio de los requisitos recopilados de los futuros usuarios y adecuados a la vista del creador del presente documento:

- a) La solución para implementar debe contar con los siguientes roles: administradores, docentes, apoderados, inspectoría, dirección.
- b) La solución para implementar debe permitir a los usuarios administradores, gestionar a todos los usuarios del sistema.
- c) El registro de los usuarios del sistema debe ser realizado solo por el rol administrador custodiado por la administración del colegio.
- d) La solución para implementar debe enviar un mail con el acceso al usuario cuando se cree una cuenta nueva.
- e) La solución para implementar debe permitir a los usuarios con los roles: administrador, docente, inspectoría y dirección poder enviar mensajes a los usuarios con rol apoderado.
- f) La solución para implementar debe enviar confirmaciones de lectura por parte del receptor al emisor del mensaje.
- g) La solución para implementar debe mostrar un directorio de contactos según nivel académico.
- h) La solución para implementar debe ser multiplataforma.
- i) La solución para implementar debe mostrar un resumen académico de los alumnos consistente en calificaciones, asistencia y anotaciones referentes a conducta, visible a los apoderados.
- j) La solución para implementar debe mostrar un calendario de fechas importantes (ceremonias, eventos, entre otros).
- k) La solución para implementar debe mostrar el horario del alumno.
- l) La solución para implementar debe mostrar las evaluaciones calendarizadas para el alumno, las cuales son previamente estipuladas por el docente responsable.

1.6. ALTERNATIVAS GLOBALES DE SOLUCIÓN.

Para establecer una solución a las problemáticas presentadas anteriormente, es difícil apartar la vista de las aplicaciones de mensajería instantánea, ya que su masividad [7] y adopción sin contar la facilidad de uso, hacen que sean privilegiadas para las comunicaciones entre los apoderados y funcionarios del colegio. Con la vista apuntando a los sistemas de mensajería instantánea para este proyecto se consideran las siguientes alternativas disponibles en el mercado.

1.6.1. Alternativa 1: Papinotas

La primera alternativa a considerar es la App "Papinotas" [8] que es una aplicación de gestión escolar multiplataforma lanzada el año 2012. Según el análisis realizado "Papinotas" es una solución de software genérica que cuenta con las siguientes funcionalidades [8]:

- Comunicaciones al hogar.
- Revisión de calificaciones y asistencia escolar.
- Descarga de archivos y certificados.

Además de lo descrito anteriormente, esta solución de software cuenta con varios módulos adicionales los cuales son ofrecidos como servicios bajo diferentes planes contractuales entre cuales se encuentran:

- Aula virtual.
- Libro de clases digital.
- Gestión de convivencia escolar.

En conclusión, se puede establecer que la solución de software genérica "Papinotas" permite a los miembros de la comunidad educativa poder tener una comunicación efectiva, acceso a información académica de los alumnos, mediante un sistema multiplataforma.

1.6.2. Alternativa 2: Kimche

“Kimche” es una solución genérica de software de gestión escolar basada en la nube. Se presenta como una herramienta para “*simplificar los procesos pedagógicos y administrativos en los establecimientos educativos*” [9]. Según el análisis realizado a esta alternativa “Kimche” ofrece las siguientes funcionalidades:

- Big Data (basado en datos de la institución educativa)
- Libro de clases digital.
- App Familia (revisión de calificaciones y asistencia escolar)
- WhatsApp Escolar (Mensajería al hogar)

En vista de las características de esta solución, se entiende que la solución “Kimche” ofrece una solución de comunicación escolar.

1.6.3. Alternativa 3: Desarrollo a medida.

El desarrollo de un sistema de mensajería escolar a medida, en primera instancia cumpliría de forma precisa las necesidades expuestas por la comunidad de El Colegio. De esta manera se acotaría a solo a las funcionalidades requeridas por la comunidad. Por otro lado, posterior a la implantación y expirado el plazo de garantía del sistema, su operación será a cargo de la institución.

1.7. EVALUACIÓN DE ALTERNATIVAS.

Para poder implementar una solución a la problemática presentada anteriormente, se realizará una evaluación de cada alternativa presentada, haciendo hincapié en los siguientes criterios.

1.7.1. Criterios para evaluar las alternativas.

- a) **Funcionalidad:** La alternativa de solución debe satisfacer los requisitos de funcionalidad mencionados anteriormente, siendo un imperativo que la solución sea multiplataforma y de mensajería instantánea.
- b) **Usabilidad:** La solución a elegir debe tener similitud de usabilidad [10] de los sistemas de mensajería instantánea más populares (Telegram, WhatsApp), con la finalidad ulterior de evitar una capacitación para el uso de la solución.
- c) **Mantenición:** La solución debe tener un mantenimiento mínimo y con un nivel de disponibilidad superior al 90%.
- d) **Seguridad de la Información:** La solución a elegir debe resguardar la información de la comunidad escolar, además de no estar expuesto a filtraciones y en un caso ideal, que no pase por terceras partes que puedan explotar la información.
- e) **Costo:** La solución debe tener un costo adecuado al presupuesto del colegio asignado a software y que no devenga un costo de mantención post implementación superior a 10UF¹.
- f) **Puesta en marcha:** Dado el volumen de la comunidad escolar de El Colegio (más de 400 estudiantes) y considerando que cada alumno debe tener dos apoderados, lo anterior supone un registro de aproximadamente 900 usuarios en el sistema. Por lo tanto, la puesta en marcha debe ser acotada y en lo posible automatizada.

¹ Valor UF al 15-12-2024: \$38.810 CLP"

1.7.2. Ponderación de Criterios y Escala de Evaluación.

A continuación, se presenta la ponderación de criterios y escala de evaluación de las alternativas de solución, en función de los criterios presentados anteriormente.

Tabla 1-1 Ponderación de Criterios

Criterio	Ponderación
Funcionalidad:	15%
Usabilidad:	20%
Mantenimiento:	10%
Seguridad de la Información:	20%
Costo:	20%
Puesta en Marcha:	15%
TOTAL:	100%

Fuente: Elaboración propia

Tabla 1-2 Escala de Evaluación

Calificación	Concepto
0	No Aplica
1	Muy Malo
2	Malo
3	Regular
4	Bueno
5	Muy Bueno

Fuente: Elaboración Propia

1.7.3. Evaluación.

1.7.3.1. Evaluación de Alternativa 1, Papinotas.

- a) **Funcionalidad:** Se evidencia que esta solución se presenta como multiplataforma y compatible con muchos dispositivos móviles, además de presentar notificaciones Push. Desafortunadamente, la función de comunicación escolar no está disponible de forma independiente, ya que forma parte de un paquete más amplio de soluciones, por lo que se restan puntos al otorgar funcionalidades que no son requeridas al alcance de esta solución.

Ponderación 4

- b) **Usabilidad:** Se aprecia en el demo [11] del producto que es posible enviar y recibir mensajes dentro de la aplicación, además de poder recibir SMS sin ser necesario tener la aplicación en el dispositivo a usar.

Ponderación 4

- c) **Mantención:** Al realizar el análisis de evaluación no fue posible obtener información de este ítem, por lo cual no se tiene certeza del nivel de disponibilidad.

Ponderación 0

- d) **Seguridad de la Información:** Lamentablemente en este ítem no fue posible obtener información acerca de la custodia de la información que realiza la empresa tras la solución "Papinotas", pero en vista de que es una entidad de Chile se debe regir por la Ley 21096 [12] y Ley 19628 [13] que rigen la protección de datos personales.

Ponderación 3

- e) **Costo:** El costo de la solución "Papinotas" es variable de acuerdo con los módulos a usar. Considerando que es una solución pensada principalmente en libro de clases digital, gran parte del valor del servicio es atribuible a este ítem. El costo mínimo por alumno es de 0,11UF² mensuales, que en año calendario suma la cantidad de 528UF, considerando 400 alumnos.

Ponderación 1

² Valor UF al 15-12-2024: \$38.810 CLP

- f) **Puesta en marcha:** La puesta en marcha según los instructivos supone un registro manual de cada apoderado, siendo el proceso invocado por el apoderado en la solución. **Ponderación 2**

Tabla 1-3 Calificación de la alternativa 1.

Criterio	Calificación
Funcionalidad:	4
Usabilidad:	4
Mantenición:	0
Seguridad de la Información:	3
Costo:	1
Puesta en Marcha:	2

Fuente: Elaboración Propia

1.7.3.2. Evaluación de Alternativa 2, Kimche.

- a) **Funcionalidad:** En el análisis de esta alternativa para este ítem, la solución Kimche se presenta como multiplataforma, siendo su principal apuesta el uso de la aplicación WhatsApp como plataforma de su sistema de mensajería. Lo anterior hace atractivo considerando la penetración de la aplicación WhatsApp [14] en el mercado de dispositivos móviles en Chile. Además esto supondría no requerir instalación por parte de los apoderados, pero es una desventaja en caso de que el usuario no use la aplicación WhatsApp.

Ponderación 4

- b) **Usabilidad:** Se aprecia en el demo [15] del producto que es posible enviar y recibir mensajes dentro de la aplicación.

Ponderación 4

- c) **Mantenición:** Kimche indica que opera sobre servicios Cloud, precisamente en AWS (Amazon Web Services) y GCP (Google Cloud Plataform), mencionado que tienen un porcentaje de disponibilidad superior al 90%.

Ponderación 4

d) Seguridad de la Información: Al igual que en la alternativa 1, lamentablemente en este ítem no fue posible obtener información acerca de la custodia de la información que realiza la empresa tras la solución "Kimche", pero en vista de que es una entidad de Chile se debe regir por la Ley 21.096 [12] y Ley 19.628 [13] que rigen la protección de datos personales.

Ponderación 3

e) **Costo:** El costo de la solución "Kimche" es variable de acuerdo con los módulos a usar. Esta solución de gestión escolar se presenta de forma modular, siendo posible contratar los servicios de forma individual. Siendo la solución de interés la de mensajería y su costo es de 0,096UF³ por alumno de forma mensual, que en año calendario suma la cantidad de 460UF, considerando 400 alumnos.

Ponderación 1

f) **Puesta en marcha:** La puesta en marcha de esta solución requiere el registro de los números de teléfono de los usuarios por parte del administrador del sistema en el sistema de Kimche, se desconoce si este proceso puede ser automático.

Ponderación 3

Tabla 1-4 Calificación de la alternativa 2.

Criterio	Calificación
Funcionalidad:	4
Usabilidad:	4
Mantenimiento:	4
Seguridad de la Información:	3
Costo:	1
Puesta en Marcha:	3

Fuente: Elaboración Propia

³ Valor UF al 15-12-2024: \$38.810 CLP

1.7.3.3. Evaluación de Alternativa 3, Desarrollo a medida.

a) Funcionalidad: El desarrollo a medida, desde su planteamiento tiene a la funcionalidad basada en la toma de requerimientos como prioridad, en estricto rigor, plantear una solución de software a medida capturará solo lo prioritario y primordial para las necesidades de mensajería que presenta El Colegio.

Ponderación 5

b) Usabilidad: Un desarrollo a medida, mediante pruebas de adaptación, se enfocará en mapear la usabilidad mediante prototipos y estándares [10]. De esta manera se asegura que la usabilidad de la solución sea la correcta.

Ponderación 5

c) Mantención: El desarrollo a medida de una solución, plantea retos para su futura mantención. De igual manera, también se consideran soluciones para que el mantenimiento de esta solución sea acotado y bien documentado.

Ponderación 5

d) Seguridad de la Información: El desarrollo a medida de una solución de software plantea retos significativos en el ámbito de la seguridad de la información. Esto se debe principalmente a los riesgos inherentes asociados a la operación de la solución por parte del colegio. Aunque las tecnologías empleadas incluyen medidas de seguridad robustas, estas no son completamente inmunes a riesgos futuros, ya sea debido al stack tecnológico utilizado, vulnerabilidades de la solución de software fuera del periodo de garantía, vulnerabilidades en componentes de infraestructura en los que se despliegue el software (servers, contenedores, entre otros). Sin embargo, una ventaja destacable de las soluciones desarrolladas a medida es la garantía en la custodia de la información de los usuarios, lo que minimiza el riesgo de filtraciones a terceros.

Ponderación 2

e) Costo: Al realizar el análisis de oferta en el mercado, en el caso de las soluciones SaaS se evidencia un costo promedio de 0,096UF⁴ mensual por alumno. Por otro lado, costo del desarrollo de una solución de software, significa una inversión

⁴ Valor UF al 15-12-2024: \$38.810 CLP

considerable de entrada superior a las 200UF. Pero considerando un universo de 400 alumnos que tiene CL, se obtiene un costo de anual de 460,93UF. En resumen un desarrollo a medida resultaría en un sistema más económico.

Ponderación 5

- f) **Puesta en marcha:** El desarrollo a medida, considera el diseño y planeamiento detallado de una puesta en marcha, por lo cual tendría la finalidad de que este evento sea acotado y sencillo.

Ponderación 5

Tabla 1-5 Calificación de la alternativa 3

Criterio	Calificación
Funcionalidad:	5
Usabilidad:	5
Mantenición:	5
Seguridad de la Información:	2
Costo:	5
Puesta en Marcha:	5

Fuente: Elaboración Propia

Resumen de Evaluación de las alternativas.

Tabla 1-6 Resumen de Evaluación de Alternativas de Solución.

Criterio	Ponderación	Alternativa 1	Alternativa 2	Alternativa 3
Funcionalidad	15%	4	4	5
Usabilidad	20%	4	4	5
Mantenición	10%	0	4	5
Seguridad de la Información	20%	3	3	2
Costo	20%	1	1	5
Puesta en Marcha	15%	2	3	5
	Total	25%	20%	40%

Fuente: Elaboración Propia

1.8. ALTERNATIVA SELECCIONADA.

De acuerdo con la información en la Tabla 1-6 fue seleccionada la alternativa 3, por obtener la mayor ponderación según la ponderación de sus criterios. A continuación, el detalle de la alternativa seleccionada.

1.8.1. Descripción y beneficios de la alternativa seleccionada

1.8.1.1. Descripción.

La solución de mensajería escolar desarrollada por cuenta propia tiene la premisa de satisfacer de forma precisa y minimalista las necesidades presentadas en la sección anterior, dando énfasis a una funcionalidad y usabilidad probada, mantenimiento simple y el punto más importante, que sea acorde al presupuesto. Es menester que el desarrollo de esta solución por cuenta propia capture lo fundamentalmente necesario y que posterior a su implantación, mejore la experiencia comunicativa de los usuarios del colegio.

1.8.1.2. Objetivo de la alternativa seleccionada

El objetivo principal de la alternativa seleccionada contempla diseñar un sistema de mensajería instantánea Escolar Multiplataforma para el Colegio Laico de Valdivia.

1.8.1.3. Beneficios

- **Unificación de las comunicaciones:**

La implementación de la solución de mensajería establecería un canal digital formal y oficial de comunicaciones desde el colegio al hogar.

- **Facilitación de entrega de comunicados:**

La solución de mensajería contempla el registro de la totalidad de apoderados existentes en El Colegio, en consecuencia, los comunicados masivos, serán fácilmente gestionados desde la solución.

- **Prescindir de la libreta de comunicaciones:**

Al contar con un servicio de mensajería, se considera una posibilidad, que la actual libreta de comunicaciones actualmente usada por El Colegio sea innecesaria pudiendo prescindir su uso.

1.8.2. Desarrollo del plan de personal

a) Un Jefe de Proyecto (JP):

El jefe de proyecto será el responsable del proyecto en general, específicamente en el seguimiento de los plazos, la asignación de recursos y la comunicación en general de los interesados del proyecto. Además, en conjunto con el Arquitecto de software realizará las tareas de análisis de requerimientos y validación de usabilidad. Es ideal que el profesional a desempeñar tenga alguna certificación en proyectos PMP (Project Management Profesional) [16].

b) Un Arquitecto de Software (ARQ):

Las funciones del arquitecto de software primordialmente residen en el diseño de la arquitectura del sistema de mensajería escolar que comprende, el modelamiento de datos, selección de tecnologías a usar, diseño y usabilidad de la interfaz de usuario, la integración de los componentes y la gestión de la configuración.

c) Un Desarrollador Full-Stack (DEV):

El desarrollador de software full-stack se encargará de la fabricación del software necesario siguiendo las directrices y documentos entregados por el Arquitecto de Software. Deberá tener experiencia en desarrollo de interfaces de usuario con ReactJS y servicios backend.

d) Un Analista Testing (QA):

El analista Testing será el encargado de asegurar la calidad del software entregado por el desarrollador. Se encargará de diseñar, planificar y ejecutar pruebas de calidad y estar en constante retroalimentación con el desarrollador para eventuales correcciones de defectos.

e) Un Analista de Operaciones (OPER):

El analista de operaciones se encargará de la implementación y puesta en marcha del sistema de mensajería, además de entregar soporte a los usuarios del sistema según la planificación acordada.

1.8.3. Planificación temporal y de recursos por actividad

Para la ejecución del proyecto en primera instancia se considerará el modelo de proceso cascada, lo cual es apto para desarrollos que involucran requerimientos y funcionalidades bien definidas y acotadas [2]. Sin embargo, en virtud de la compartimentalización, interdependencia y validación del esfuerzo [2] especialmente en las etapas de pruebas (Testing), se recomienda un enfoque iterativo. Considerando la naturaleza del proyecto, los recursos disponibles y los requisitos planteados por los usuarios, se optará por una metodología híbrida que combine los modelos cascada e incremental.

Como resultado, la calendarización del proyecto contempla la construcción del software en distintos módulos, en línea con un enfoque incremental. Sin embargo, se realizará una única entrega final del software completo.

Además según el análisis del estado del arte en soluciones y librerías de software similares a la planteada en este proyecto, se considera el uso de librerías y componentes FOSS(Free Open Source Software) con un gran soporte y documentación, tales como el protocolo de mensajería XMPP [17] y la librería ReactJS [18]. Con el uso de estas tecnologías, se espera acortar el ciclo de construcción y diseño, por lo cual la metodología híbrida mencionada anteriormente, es adecuada al considerar estas tecnologías.

En la siguiente tabla 1-7 se presenta la planificación de las actividades del proyecto sumariadas en una carta Gantt, considerando un inicio tentativo del proyecto para el día 31 de marzo de 2025.

Tabla 1-7 Carta Gantt

Id	Nombre de tarea	Duración (Días)	Comienzo	Fin	Actividad Predecesora	Recurso
1	Proyecto Sistema de Mensajería Escolar	51	Mon 31-03-25	Mon 09-06-25		
2	Firma de Contrato	0,5	Mon 31-03-25	Mon 31-03-25		JP
3	KickOFF	0,5	Mon 31-03-25	Mon 31-03-25	2	JP, ARQ
4	Modulo Mensajería	15	Tue 01-04-25	Thu 17-04-25	3	
5	Análisis de Requerimientos	3	Tue 01-04-25	Thu 03-04-25		JP, ARQ
6	Diseño	2	Tue 01-04-25	Tue 01-04-25	5	JP (50%), ARQ
7	Construcción	7	Wed 02-04-25	Mon 14-04-25	6	
8	Base de Datos	1	Wed 02-04-25	Wed 02-04-25		DEV, ARQ
9	Aplicaciones Interactivas	2	Wed 02-04-25	Tue 08-04-25	8	JP (50%),DEV
10	Procesos Batch	1	Wed 09-04-25	Wed 09-04-25	8	DEV
11	Interfaz - Gui	2	Thu 10-04-25	Fri 11-04-25	10	JP (50%),DEV
12	Fin de Construcción	1	Mon 14-04-25	Mon 14-04-25	11	DEV,ARQ,JP
13	Pruebas - Testing	3	Tue 15-04-25	Thu 17-04-25	12	QA
14	Modulo Indicadores Escolares	14	Fri 18-04-25	Wed 07-05-25	13	
15	Análisis de Requerimientos	3	Fri 18-04-25	Tue 22-04-25		JP, ARQ
16	Diseño	1	Wed 23-04-25	Wed 23-04-25	15	JP (50%),ARQ
17	Construcción	7	Thu 24-04-25	Fri 02-05-25	16	
18	Base de Datos	1	Thu 24-04-25	Thu 24-04-25	16	JP (50%),DEV, ARQ
19	Aplicaciones Interactivas	2	Thu 24-04-25	Mon 28-04-25	18	JP (50%),DEV
20	Procesos Batch	1	Mon 28-04-25	Tue 29-04-25	18	DEV
21	Interfaz - Gui	2	Tue 29-04-25	Thu 01-05-25	20	JP (50%),DEV
22	Fin de Construcción	1	Thu 01-05-25	Fri 02-05-25	21	DEV,ARQ,JP
23	Pruebas - Testing	3	Fri 02-05-25	Wed 07-05-25	22	QA
24	Mantenedores	14	Wed 07-05-25	Thu 29-05-25	14	
25	Análisis de Requerimientos	3	Wed 07-05-25	Mon 12-05-25		JP, ARQ
26	Diseño	1	Mon 12-05-25	Tue 13-05-25	25	ARQ
27	Construcción	7	Tue 13-05-25	Mon 26-05-25	26	
28	Base de Datos	1	Tue 13-05-25	Tue 13-05-25	26	DEV, ARQ
29	Aplicaciones Interactivas	2	Wed 14-05-25	Tue 20-05-25	28	JP (50%),DEV
30	Procesos Batch	1	Tue 20-05-25	Wed 21-05-25	28	DEV
31	Interfaz - Gui	2	Wed 21-05-25	Fri 23-05-25	30	JP (50%),DEV
32	Fin de Construcción	1	Fri 23-05-25	Mon 26-05-25	31	DEV,ARQ,JP
33	Pruebas - Testing	3	Mon 26-05-25	Thu 29-05-25	32	QA
34	Implementación y Puesta en Marcha	6,5	Thu 29-05-25	Fri 06-06-25	24	
35	Planificación e implementación de piloto	0,5	Thu 29-05-25	Thu 29-05-25		JP - OPER
36	Puesta en marcha	1	Fri 30-05-25	Fri 30-05-25	35	OPER - JP
37	Entrega de soporte	5	Mon 02-06-25	Fri 06-06-25	36	JP (50%),OPER
38	Cierre del Proyecto	0,5	Mon 09-06-25	Mon 09-06-25	37	JP

Siglas: JP: Jefe de Proyecto | ARQ: Arq. De software | DEV: Desarrollador full Stack

QA: Analista QA | OPER: Analista Operaciones

Fuente: Elaboración propia

1.8.4. Administración del riesgo

Considerando que los riesgos son eventos inciertos que impactan en un proyecto [16] para el presente proyecto se considera la gestión del riesgo, siendo este elemento parte esencial de un proyecto de software exitoso.

A continuación, se mostrará la fórmula que entrega el orden de prioridad de los riesgos según su naturaleza y probabilidad.

$$prioridad = (11 - posibilidad) * (11 - impacto) * costo de retiro$$

- a) **Prioridad:** Mientras el valor sea más cercano a 1, es más alta la prioridad del riesgo.
- b) **Posibilidad:** Corresponde a la probabilidad de ocurrencia del riesgo, siendo 1 el valor menor y 10 el valor máximo.
- c) **Impacto:** Corresponde al efecto que un riesgo puede producir, el valor es entre 1 y 10, siendo el menor valor poco impacto y el mayor valor impacto grave.
- d) **Costo de Retiro:** Consiste en el costo monetario de mitigar un riesgo. Este valor transcurre en una escala de 1, menor costo a 10, que corresponde a mayor costo.

1.8.4.1. Riesgo del Proyecto

La ejecución del presente proyecto contempla riesgos en cada etapa de planeamiento y construcción, por ende es menester recapitular y considerar los riesgos comunes que se pueden presentar, A continuación se presentará una matriz de los riesgos identificados (Tabla 1-8) junto con el plan de mitigación (Tabla 1-9 | Tabla 1-10) para cada riesgo presentado:

Tabla 1-8 Administración de riesgos

N°	Ítem	Riesgo	Posibilidad	Impacto	Costo de Retiro	Prioridad
			Posibilidad (1 a 10) 1 = Menor			
1	Proyecto	Incumplimiento de los plazos establecidos	5	7	2	48
2	Producto de Software	Incumplimiento de las expectativas de los usuarios	5	9	6	72
3	Tecnología	Incompatibilidad con los dispositivos de usuarios en uso de tecnología multiplataforma.	2	5	2	108
4	Proyecto	Estimación incorrecta de los recursos	5	8	8	144
5	Proyecto	Problemas de soporte y capacitación	4	4	3	147
6	Producto de Software	Requisitos no claros y/o incompletos	5	7	7	168
7	Tecnología – Producto de Software	Posible brecha de seguridad y consiguiente filtración de información privada de los usuarios.	5	7	7	168
8	Proyecto	Cambios en los Requisitos	2	8	7	189
9	Producto de Software	Control de calidad inadecuado	2	5	5	270
10	Proyecto	Errores de implementación	2	2	4	324

Fuente: Elaboración Propia

Tabla 1-9 Mitigación de Riesgos - Parte 1

N°	Prioridad	Señal de Alerta	Plan de Mitigación		Responsable
			Acciones Preventivas	Acciones Reactivas	
1	48	Retraso en las entregas e hitos del proyecto.	<ul style="list-style-type: none"> • Establecer plazos holgados y realistas. • Manejar planes de contingencia ante eventuales retrasos. 	<ul style="list-style-type: none"> • Entrega de nuevo plazo de entrega para el lanzamiento de la solución. 	JP
2	72	Clientes no satisfechos con la funcionalidad del aplicativo.	<ul style="list-style-type: none"> • Realizar pruebas de aceptación y encuestas de satisfacción con los usuarios, mediante el uso de sw similar y prototipos, antes de comenzar el desarrollo. 	<ul style="list-style-type: none"> • Crear un proceso de mejoramiento de expectativas de los usuarios, recopilando los datos de insatisfacción y generando mejoras para obtener un porcentaje de aprobación superior al 75%. 	JP - ARQ
3	108	Usuarios con dispositivos legacy no pueden usar el aplicativo.	<ul style="list-style-type: none"> • Comprobar las versiones de la tecnología a usar y que sea compatible con la mayoría de los dispositivos. • Realizar un breve levantamiento de los dispositivos móviles que tienen los usuarios 	<ul style="list-style-type: none"> • Algunos dispositivos legacy lamentablemente no podrán ser integrados, por lo cual a aquellos usuarios se instará a usar el aplicativo versión web 	DEV-OPER-JP
4	144	Falta de recursos, balance de caja incongruente con lo proyectado.	<ul style="list-style-type: none"> • Establecer un proceso de revisión de la estimación de recursos a intervalos regulares. 	<ul style="list-style-type: none"> • Revisión de los costos asociados del proyecto con una consecuente nueva entrega de recursos para continuar con la implementación de la solución. 	JP
5	147	Usuarios con leve adherencia al aplicativo.	<ul style="list-style-type: none"> • Documentar el software de forma clara y concisa. • Crear la documentación necesaria para la autocapacitación de los usuarios. 	<ul style="list-style-type: none"> • Considerar recapacitación al personal encargado de soporte, además de ser necesario ejecutar más sesiones de capacitación con los usuarios. 	OPER
6	168	Se presentan problemas de comprensión de las funcionalidades.	<ul style="list-style-type: none"> • Obtener la aprobación de los requerimientos por parte del cliente (sostenedor colegio) antes de comenzar el desarrollo. • Realizar un análisis exhaustivo de los requisitos al principio del proyecto. • Documentación y especificación concisa de los requisitos, además de entregar un plazo holgado para esta tarea. 	<ul style="list-style-type: none"> • Elaborar un plan para completar los requisitos, ya sean claros o incompletos 	JP - ARQ

Siglas: **JP:** Jefe de Proyecto | **ARQ:** Arq. De software | **DEV:** Desarrollador full Stack

QA: Analista QA | **OPER:** Analista Operaciones

Fuente: Elaboración propia

Tabla 1-10 Mitigación de Riesgos - Parte 2

N°	Prioridad	Señal de Alerta	Plan de Mitigación		Responsable
			Acciones Preventivas	Acciones Reactivas	
6	168	Se presentan problemas de compresión de las funcionalidades.	<ul style="list-style-type: none"> • Obtener la aprobación de los requerimientos por parte del cliente (sostenedor colegio) antes de comenzar el desarrollo. • Realizar un análisis exhaustivo de los requisitos al principio del proyecto. • Documentación y especificación concisa de los requisitos, además de entregar un plazo holgado para esta tarea. 	<ul style="list-style-type: none"> •Elaborar un plan para completar los requisitos, ya sean claros o incompletos 	JP – ARQ
7	168	Actividad inusual en el tráfico normal en el aplicativo	<ul style="list-style-type: none"> • Usar protocolos y esquemas que disminuyan las brechas de seguridad ya sean en el desarrollo del software como en la operación. 	<ul style="list-style-type: none"> • En caso de brecha de seguridad se contará con un plan de mitigación de seguridad que se ejecute bajo los siguientes escenarios: ransomware, data breach, ddos. 	DEV-OPER-JP
8	189	Cientes proponen nuevas funcionalidades y/o ideas que cambian el alcance del proyecto, una vez finalizado la toma de requerimientos.	<ul style="list-style-type: none"> •Establecer un proceso formal con firma de un documento, para la gestión de los cambios de requisitos. •Evaluar el impacto de los cambios de requisitos en el cronograma, el presupuesto y los recursos del proyecto, en caso de que el cambio sea irreductible. •Establecer expectativas claras con el cliente (sostenedor colegio) sobre el eventual cambio de los requisitos. 	<ul style="list-style-type: none"> •Establecer un ajuste en el cronograma de trabajo propuesto para hacer frente al cambio de requisitos. 	JP – ARQ
9	270	Errores de funcionalidad / funcionalidades incompletas en aplicativo.	<ul style="list-style-type: none"> • Involucrar a los usuarios en el proceso de control de calidad. 	<ul style="list-style-type: none"> • Revisión y corrección de defectos de desarrollo, por el equipo correspondiente. 	DEV – QA
10	324	Imprevistos en la puesta en marcha.	<ul style="list-style-type: none"> • Realizar pruebas de aceptación y encuestas de satisfacción con los usuarios, mediante el uso de prototipos, antes de comenzar el desarrollo. 	<ul style="list-style-type: none"> • Uso de protocolo de manejo de errores de implementación, con la finalidad de sortear la implementación pese a los errores presentados. 	JP – OPER

Siglas: **JP:** Jefe de Proyecto | **ARQ:** Arq. De software | **DEV:** Desarrollador full Stack

QA: Analista QA | **OPER:** Analista Operaciones

Fuente: Elaboración propia

1.8.5. Estimación de costos

Considerando el alcance del proyecto que está acotado a un sistema de mensajería e indicadores, para la fabricación de esta solución se usará la metodología de cascada [2], considerando que hay una visión clara del producto final, el alcance del proyecto está bien definido y como premisa se utilizarán piezas de software FOSS ya desarrolladas para acometer la solución de software.

En base a lo anterior se presenta a continuación el detalle de los recursos humanos y sus costos mensuales (Tabla 1-11) involucrados en la ejecución del proyecto, considerando 20 días laborales por mes y cada jornada de ocho horas diarias.

Para obtener la glosa de remuneraciones de los distintos profesionales para el proyecto, se consultó la referencia de salarios promedio del sitio Talent.com [19].

Se ilustran además los costos fijos mensuales (Tabla 1-12) los que incluyen el arriendo de oficinas tipo cowork para gestionar el trabajo, servicios básicos de oficina, servicios de internet para el personal y otros costos asociados tales como viáticos, almuerzos e imprevistos que pueden surgir durante la ejecución del proyecto.

Tabla 1-11 Costo de Recursos

Recurso	Sueldo Mensual UF⁵	% del total
Jefe de Proyecto	80	29%
Arquitecto de SW	70	25%
Desarrollador	60	21%
Analista Testing	35	13%
Analista Operaciones	32	12%
TOTAL	277 UF	100%

Fuente: Talent Chile [19]

⁵ Valor UF al 15-12-2024: \$38.810 CLP

Tabla 1-12 Costos de Servicios

Costo Fijo	Costo UF
Arriendo	30
Servicios Básicos	2
Comunicaciones	3
Otros Costos	10
TOTAL	45 UF

Fuente: Elaboración propia

La ejecución del proyecto considera un recurso humano encargado de la venta y tareas administrativas ajenas a la gestión de software, por ende se contará con un consultor especializado en venta con experiencia en educación, el costo del consultor se expresa en la Tabla 1-13.

Tabla 1-13 Costos de Venta

Gasto de Venta	Costo UF⁶
Consultor	35
Comunicaciones	1
Viáticos	2
TOTAL	38 UF

Fuente: Elaboración propia

⁶ Valor UF al 15-12-2024: \$38.810 CLP

Tabla 1-14 Valor Costo Empresa

Costos	Costo UF
Costo RR.HH.	277
Costos Fijos - Servicios	45
Costo Venta	38
COSTO EMPRESA MENSUAL	360 UF

Fuente: Elaboración propia

De la Tabla 1-14 se desprende que el costo empresa es de 360 UF, esto como resultado de la suma entre los siguientes elementos: el costo de personal, los gastos fijos relativos a servicios y el pago al consultor encargado de la venta.

Continuando con estimación del costo total del proyecto, es necesario realizar un prorrateo del costo empresa en relación con el valor del esfuerzo de cada profesional el cual se presenta en la Tabla

Tabla 1-15 Costo de recursos ajustados

RR.HH.	Costo RR.HH. mes	% del total	Costo RR.HH. Prorrateado	Costo RR.HH. diario (20 días)	Días proyecto	Total RR.HH. Proyecto
JP	80	29%	104,0	5,2	25,5	132,6
ARQ	70	25%	91,0	4,5	19,5	88,7
DEV	60	21%	78,0	3,9	21	81,9
QA	35	13%	45,5	2,3	9	20,5
OPER	32	12%	41,6	2,1	6,5	13,5
TOTAL	277 UF	100%	-	Costo Empresa Proyecto:		337,2

Siglas: **JP:** Jefe de Proyecto | **ARQ:** Arq. De software | **DEV:** Desarrollador full Stack

QA: Analista QA | **OPER:** Analista Operaciones

Fuente: Elaboración propia

De la Tabla 1-15 se desprende que el costo total del proyecto es 337,2 UF basado en el esfuerzo necesitado por profesional para ejecutar el proyecto.

Según las reglas de negocio es necesario agregar al costo del proyecto se agregará un porcentaje por concepto de riesgo, en caso de imprevistos relacionados con el financiamiento de créditos, este ítem corresponderá al 5% del valor del proyecto. Se ilustra lo anterior en la Tabla 1-16

Tabla 1-16 Precio total del proyecto

Descripción	Total UF
Total costo Proyecto	337,20
Riesgo 5 %	16,86
Precio Total del Proyecto	354,6

Fuente: Elaboración propia

Considerando lo anterior se desprende que el precio total del proyecto, el cual pagará el cliente corresponde a 354,6 unidades de fomento.

**2. CAPÍTULO 2: ASPECTOS RELEVANTES DEL ANÁLISIS PARA LA
ALTERNATIVA SELECCIONADA**

2.1. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

La alternativa seleccionada consiste en el desarrollo de una solución de software propia, que tiene la finalidad de flanquear las problemáticas presentadas en el capítulo anterior, tales como: unificar las comunicaciones escolares, prescindir de la libreta de papel, entre otras. De esta manera El Colegio Laico podrá centralizar las necesidades de comunicación y disponibilidad de información para la comunidad educativa.

2.1.1. Funcionalidades de la solución

A continuación, se presenta un resumen de los elementos necesarios de la solución de software:

2.1.1.1. Roles

La solución debe contar con los siguientes roles de usuario basados en el análisis de los funcionarios de El Colegio laico:

- **Administrador:** Corresponde al rol *superusuario*, el cual tendrá privilegios para realizar tareas administrativas y de mantenimiento de los todos los usuarios del sistema.
- **Apoderado:** Este rol está destinado para los padres y apoderados del colegio.
- **Docente:** Este rol está destinado para los docentes del colegio.
- **Funcionario:** Este rol está destinado para los funcionarios del colegio tales como: Paradocentes y profesionales del aprendizaje (psicólogos, psicopedagogos)

2.1.1.2. Módulos y/o vistas.

Esta solución de software contempla los siguientes módulos:

- **Módulo de mensajería:** Donde los usuarios pueden redactar y ver mensajes. En el caso de los usuarios con rol *apoderado*, para enviar un mensaje, este debe ser moderado de forma asíncrona por un usuario *administrador*, el cual dará el visto bueno enviándolo al destinatario o en caso contrario rechazando la solicitud.
- **Módulo de comunicados:** En este ítem los usuarios pueden ver los comunicados generales emitidos por el colegio. También los usuarios con permisos pueden redactar un comunicado general.
- **Módulo de indicadores:** En esta vista los usuarios con rol *apoderado* pueden ver los indicadores escolares (notas, anotaciones conductuales) de sus pupilos asignados en sistema. Los demás usuarios del sistema (*administrador*, *docente*, *funcionario*) pueden registrar estos indicadores y/o modificar registros existentes.
- **Módulo mantenedor:** En este módulo los usuarios administradores realizan las tareas administrativas de registro, eliminación, bloqueo de todos los usuarios.
- **Módulo horario:** El módulo horario está pensado para que los usuarios *apoderados* puedan ver el horario de clases de los pupilos registrados en su cuenta.
- **Módulo agenda Eventos:** La funcionalidad de este módulo reside en la visualización de los *eventos generales* tales como "día del profesor", vacaciones, entre otros; además de los *eventos personales* correspondientes a entrevistas de profesores y funcionarios con determinados *apoderados*. Los usuarios con privilegios correspondientes pueden crear estos y citaciones, en el caso de los usuarios con rol *Apoderado* esta citación a entrevista está mediada por un usuario de rol *Admin* quien verificará la solicitud antes de cursarla al destinatario correspondiente.

2.2. DIAGRAMAS DE ACTIVIDADES GENERAL

A continuación, se presentarán los diagramas más relevantes por actor. Cada diagrama ilustra las funcionalidades mapeadas, producto del análisis, los cuales se encuentran relacionados con los objetivos de estudio que se requiere lograr con la implementación de la solución propuesta. Cabe destacar que algunas funcionalidades por actor se omitieron de ser representadas en los siguientes diagramas, con la finalidad de evitar redundancia.

2.2.1. Flujo Administrativo Usuario Apoderado:

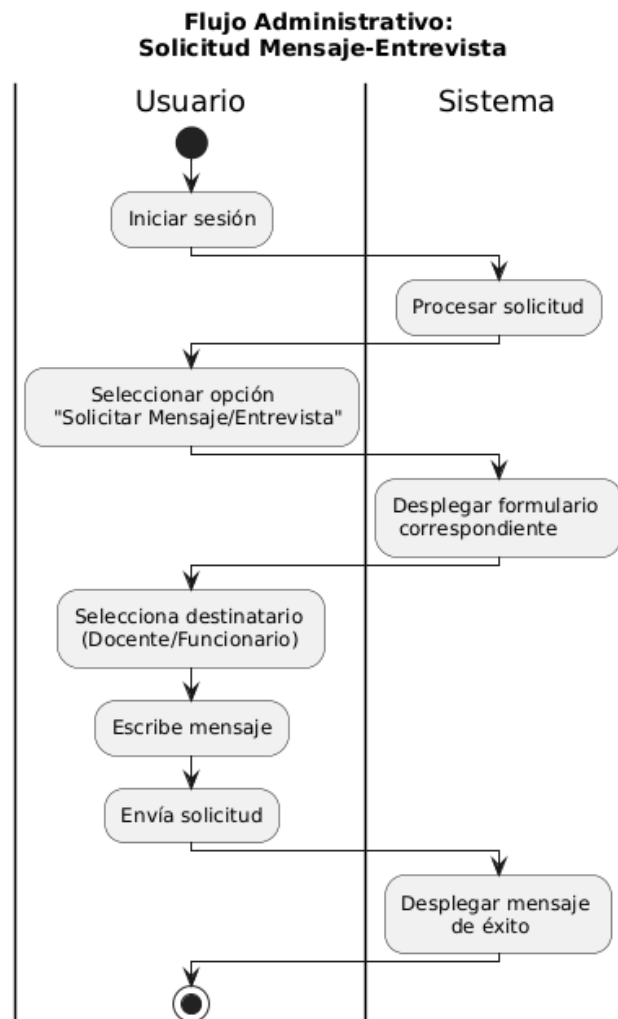


Ilustración 2-2 Flujo administrativo Actor Apoderado
Fuente: Elaboración Propia

2.2.2. Flujos Administrativos Usuario Docente

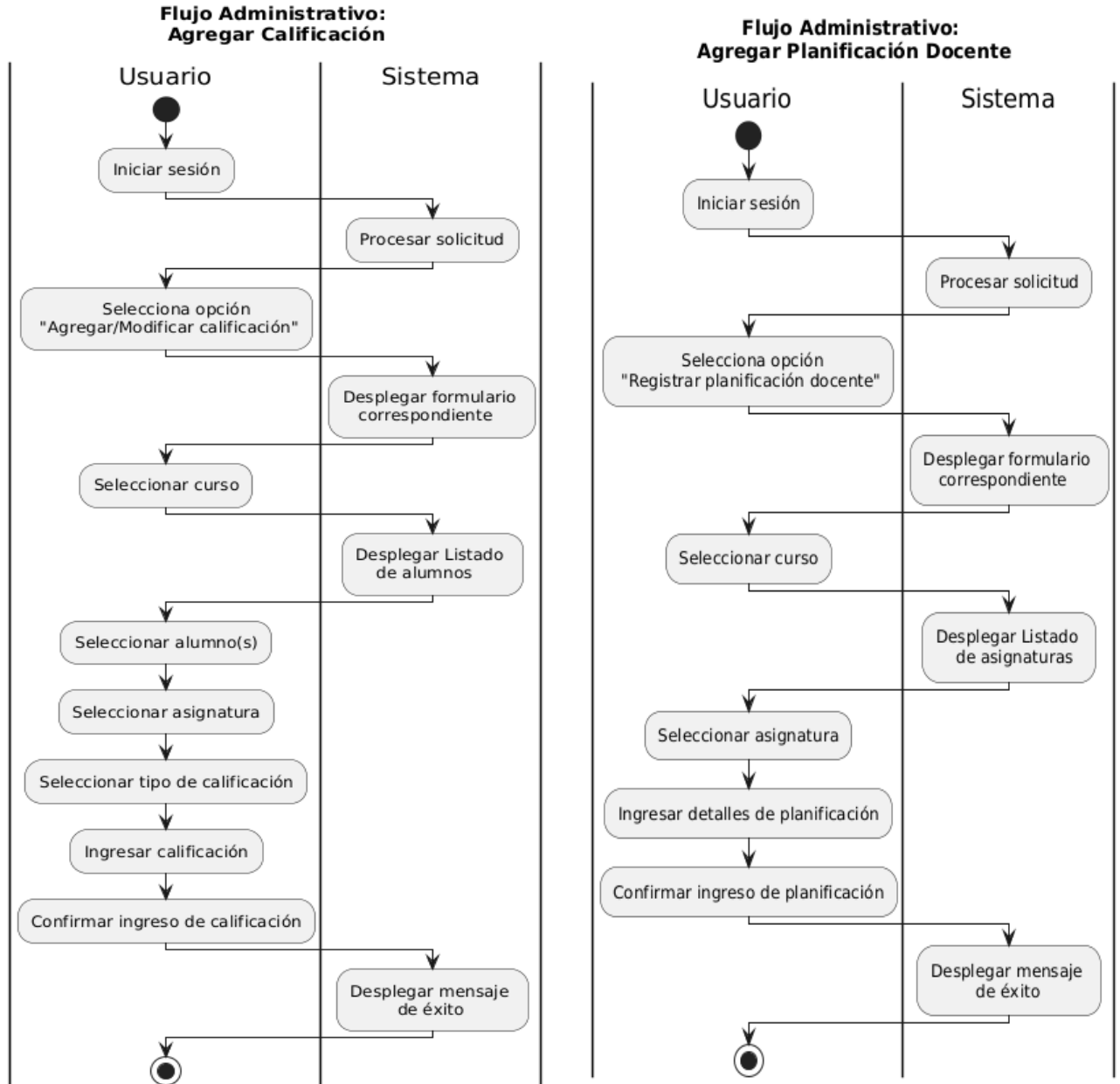


Ilustración 2-3 Flujos administrativo Actor Docente
Fuente: Elaboración Propia

2.2.3. Flujo Administrativo Usuario Funcionario

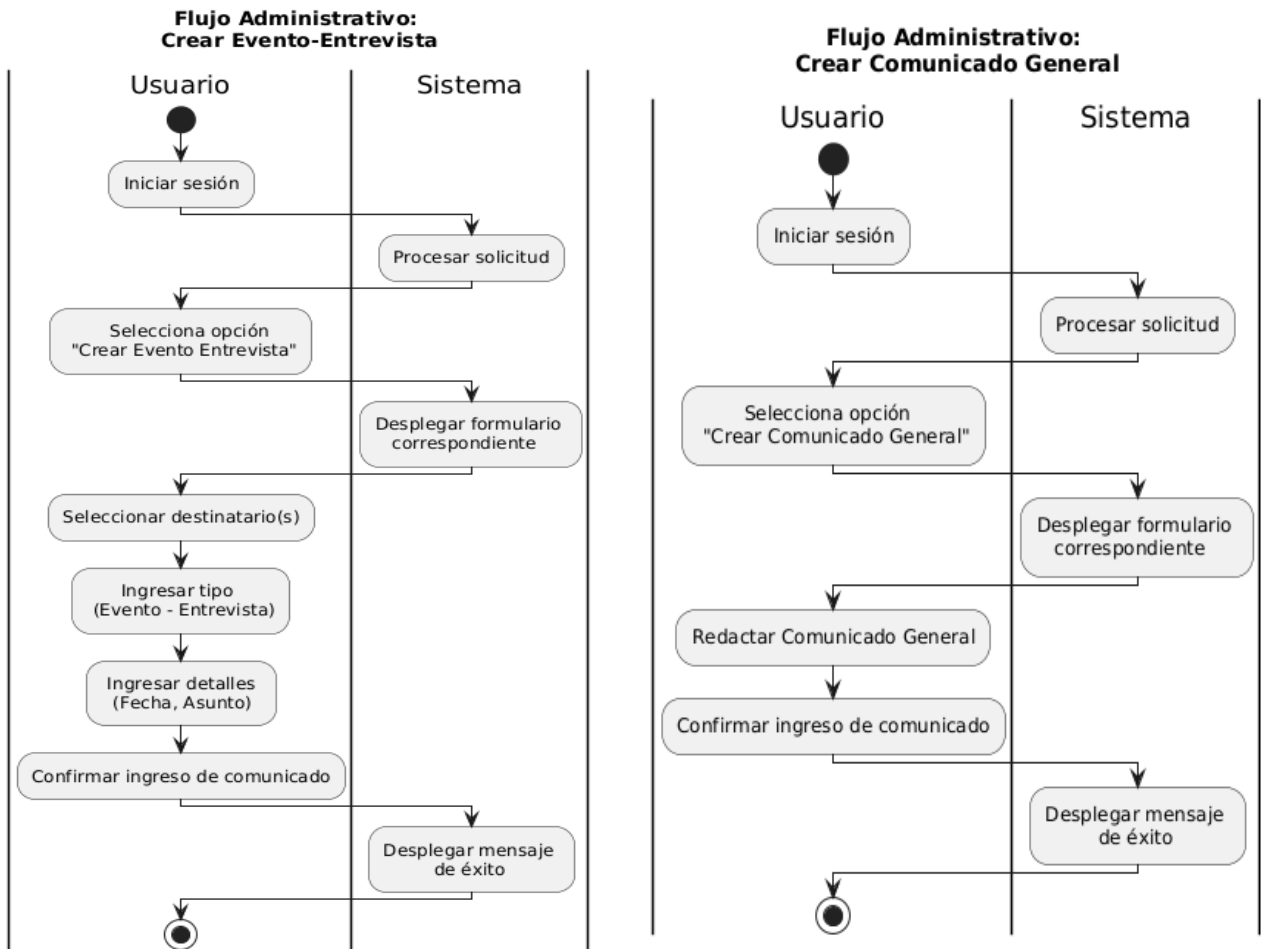


Ilustración 2-4 Flujos administrativo Actor Funcionario
Fuente: Elaboración Propia

2.2.4. Flujo Administrativo Usuario Administrador (Admin)

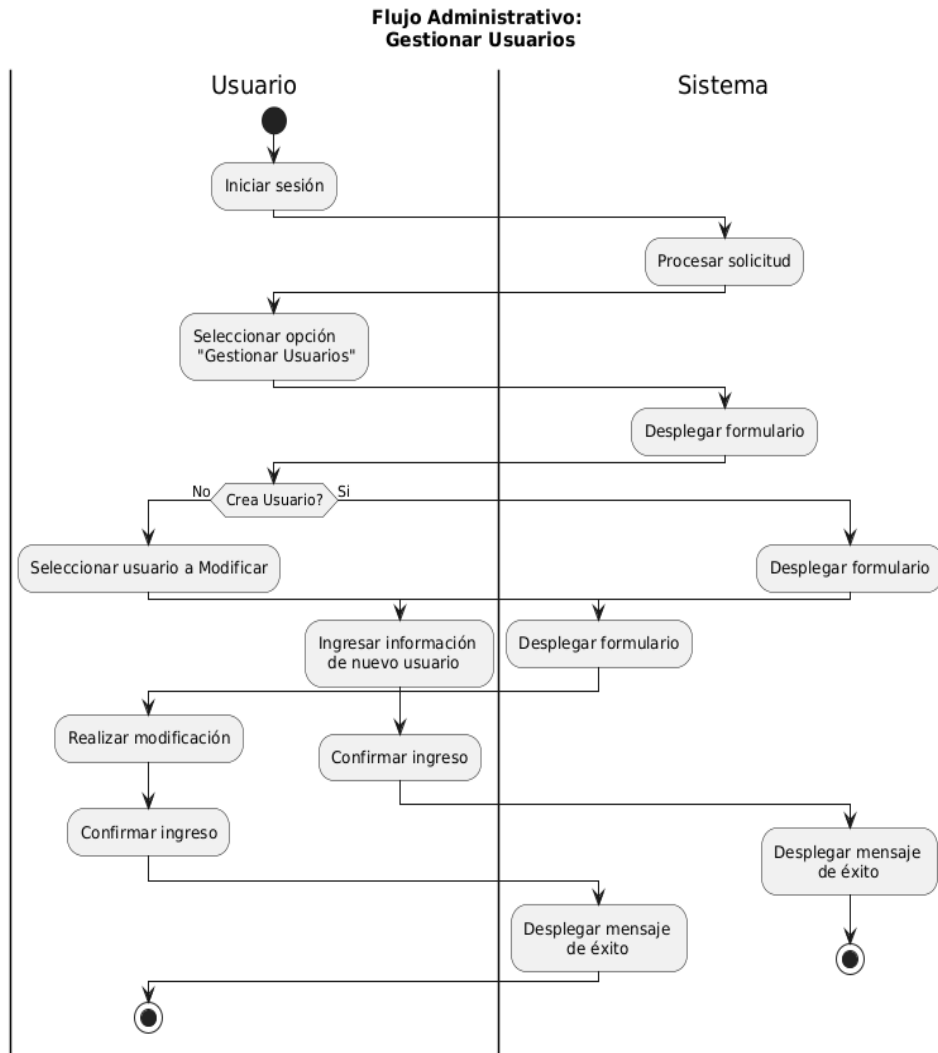


Ilustración 2-5 Flujo administrativo Actor Administrador
Fuente: Elaboración Propia

2.3. REQUERIMIENTOS DEL SISTEMA

A continuación, se especificarán los requerimientos (Tabla 2-17) o necesidades del usuario descritos en el capítulo uno, obtenidos mediante técnicas de ingeniería de requerimientos las cuales apuntan a facilitar la definición de un alcance en conjunto con el cliente [2] y un buen índice de satisfacción con la solución de software. Las especificaciones de los requerimientos serán clasificados en requerimientos funcionales, que corresponden a los servicios que debe proporcionar el sistema y requerimientos no funcionales que corresponden a las restricciones del sistema y/o efecto de calidad [20].

2.3.1. REQUERIMIENTOS FUNCIONALES

Tabla 2-17 Requerimientos Funcionales

ID	Descripción Requerimiento
RF-01	El sistema debe permitir a los usuarios con cualquier rol ver mensajes enviados a su casilla
RF-02	El sistema debe permitir a los usuarios con rol Apoderado solicitar <i>envío de mensaje</i> o <i>entrevista</i> a los usuarios con rol Docente, Funcionario y Admin. Esta solicitud es limitada a solo una solicitud por destinatario, lo anterior debido a que el mensaje debe pasar por el proceso de moderación por un usuario con rol Administrador.
RF-03	El sistema debe permitir a los usuarios con rol Apoderado ver los horarios de sus pupilos asignados.
RF-04	El sistema debe permitir a los usuarios con rol Apoderado ver los indicadores escolares (notas, anotaciones de conducta) de sus pupilos asignados.
RF-05	El sistema debe permitir a los usuarios con rol Apoderado ver la agenda correspondiente de sus pupilos asignados.
RF-06	El sistema debe permitir a los usuarios con rol Apoderado, Docente, Funcionario y Admin ver los comunicados generales.
RF-07	El sistema debe permitir a los usuarios con rol Docente, Funcionario y Admin enviar mensajes a los usuarios con rol Apoderado.
RF-08	El sistema debe permitir a los usuarios con rol Docente, Funcionario y Admin crear eventos
RF-09	El sistema debe permitir a los usuarios con rol Docente, Funcionario y Admin citar a entrevista a los usuarios con rol Apoderado.
RF-10	El sistema debe permitir a los usuarios con rol Docente y Admin agregar anotaciones de conducta a los alumnos asignados a su perfil.
RF-11	El sistema debe permitir a los usuarios con rol Docente y Admin agregar/modificar notas a los alumnos asignados a su perfil.
RF-12	El sistema debe permitir a los usuarios con rol Docente y Admin registrar las planificaciones docentes asignadas según su perfil.
RF-13	El sistema debe permitir a los usuarios con rol Funcionario y Admin la creación de comunicados generales.
RF-14	El sistema debe permitir a los usuarios con rol Funcionario y Admin la modificación de los eventos ya creados en el calendario.
RF-15	El sistema debe permitir a los usuarios con rol Admin el registro, activación, suspensión y eliminación de los usuarios con rol Apoderado, Docente y Funcionario.
RF-16	El sistema debe permitir a los usuarios con rol Admin, revisar y moderar los <i>mensajes y solicitudes de entrevista</i> enviados por los usuarios con rol apoderado, con el fin de autorizar su envío a los destinatarios.
RF-17	El sistema debe permitir a los usuarios con rol Admin cargar y modificar los horarios de clases.

Fuente: Elaboración Propia

2.3.2. REQUERIMIENTOS NO FUNCIONALES

Tabla 2-18 Requerimientos No Funcionales

ID	Descripción Requerimiento	Verificación Requerimiento
RNF-01	El acceso al sistema debe ser restringido.	El sistema solo permitirá el ingreso a los usuarios previamente registrados por un usuario Admin y con estado activo.
RNF-02	El sistema debe ser multiplataforma siendo posible utilizar en Android, IO'S y PC	Se usará la tecnología React Native que permite hacer despliegues nativos multiplataforma.
RNF-03	El sistema debe soportar concurrencia mínima de 300 usuarios.	En la etapa de Testing se realizarán pruebas de carga, donde se simulará un tráfico de 300 usuarios concurrentes. Además se contempla un plan de recuperación antes desastres para mantener el nivel de concurrencia propuesto.

Fuente: Elaboración Propia

2.4. ESTRUCTURA FUNCIONAL DEL SISTEMA - CASO DE USO GENERAL

En la presente sección se desarrollan e ilustrarán mediante técnicas de ingeniería, los casos de uso de la solución de software, identificando los actores y sus interacciones con cada funcionalidad del sistema.

2.4.1. Casos de Uso

Tabla 2-19 Casos de Uso

Código CU	Nombre Caso de Uso
CU-01	Enviar Mensaje a apoderado
CU-02	Crear evento - Entrevista apoderado
CU-03	Agregar - Modificar nota
CU-04	Agregar - Modificar anotación conductual
CU-05	Registrar planificación docente
CU-06	Ver comunicados generales
CU-07	Crear comunicados
CU-08	Modificar Calendario Eventos
CU-09	Mantener usuarios
CU-10	Ver mensajes
CU-11	Solicitar Mensaje / Entrevista
CU-12	Ver horario Alumno
CU-13	Ver indicadores
CU-14	Ver Agenda
CU-15	Moderar solicitudes de apoderados
CU-16	Cargar Horarios

Fuente: Elaboración Propia

2.4.2. Actores del sistema

2.4.2.1. Administrador:

Este actor le corresponde el rol de super usuario o root, es decir puede realizar todas las acciones disponibles del sistema. Enumerando las tareas de este actor se cuenta con las siguientes funcionalidades: encargado de la mantención del sistema en cuanto al registro, eliminación y suspensión de usuarios; moderación de mensajes; creación y edición de eventos; creación y edición de comunicados generales, creación y edición de agendas. En conclusión este rol está reservado para los encargados de informática del colegio Laico los cuales actualmente realizan tareas administrativas informáticas que un tenor similar.

2.4.2.2. Apoderado:

Este actor está reservado para los padres y apoderados del colegio, los cuales son los adultos responsables de los alumnos con matrícula vigente. Su espacio de usuario permite las siguientes funcionalidades: ver mensajes enviados, solicitar mensaje y/o entrevista, ver horario del alumno, ver comunicados generales, ver agenda y ver indicadores escolares los cuales corresponden a notas y anotaciones de conducta.

2.4.2.3. Docente:

El actor docente, que en el sistema se representa por el rol homónimo {docente}, corresponde a los profesores de El Colegio Laico que actualmente está impartiendo clases lectivas y/o talleres. Los actores del tipo docente pueden fungir las siguientes funcionalidades en el sistema: enviar mensajes, crear eventos y/o citas, agregar nota a un alumno, agregar anotación de conducta, registro de planificación docente.

2.4.2.4. Funcionario:

El actor de tipo funcionario, corresponde a los siguientes funcionarios que se desempeñan en El Colegio Laico: Inspectores, Inspector General, Psicóloga escolar, encargada programa PIE (integración escolar), encargada de convivencia. Las funcionalidades de este actor/rol son: envío de mensajes, creación de comunicados generales, modificación de calendario, creación de eventos.

2.4.3. Caso de uso General.



Ilustración 2-6 Diagrama de Casos de Uso General
Fuente: Elaboración Propia

2.5. MATRIZ DE TRAZABILIDAD

En conjunción con la definición completa de los casos de uso y los requerimientos funcionales establecidos posterior al análisis correspondiente, se presenta a continuación la matriz de trazabilidad [20]. Esta matriz tiene como objetivo principal facilitar la planificación, gestión y control del proyecto, sus tareas y actividades asociadas. Adicionalmente, permite verificar la correspondencia entre cada requerimiento funcional y al menos un caso de uso, y viceversa.

Tabla 2-20 Matriz de Trazabilidad

		Casos de Uso															
		CU 01	CU 02	CU 03	CU 04	CU 05	CU 06	CU 07	CU 08	CU 09	CU 10	CU 11	CU 12	CU 13	CU 14	CU 15	CU 16
Requerimientos Funcionales	RF 01										■						
	RF 02											■					
	RF 03												■				
	RF 04													■			
	RF 05														■		
	RF 06						■										
	RF 07	■															
	RF 08		■														
	RF 09		■														
	RF 10				■												
	RF 11			■													
	RF 12					■											
	RF 13							■									
	RF 14								■								
	RF 15									■							
	RF 16															■	
	RF 17																■

Fuente: Elaboración Propia

2.6. MODELO CONCEPTUAL

A continuación, se presenta el modelo conceptual del sistema, el cual detalla las asociaciones y relaciones de los conceptos clave necesarios para cumplir con los requerimientos funcionales establecidos en los casos de uso previamente definidos.

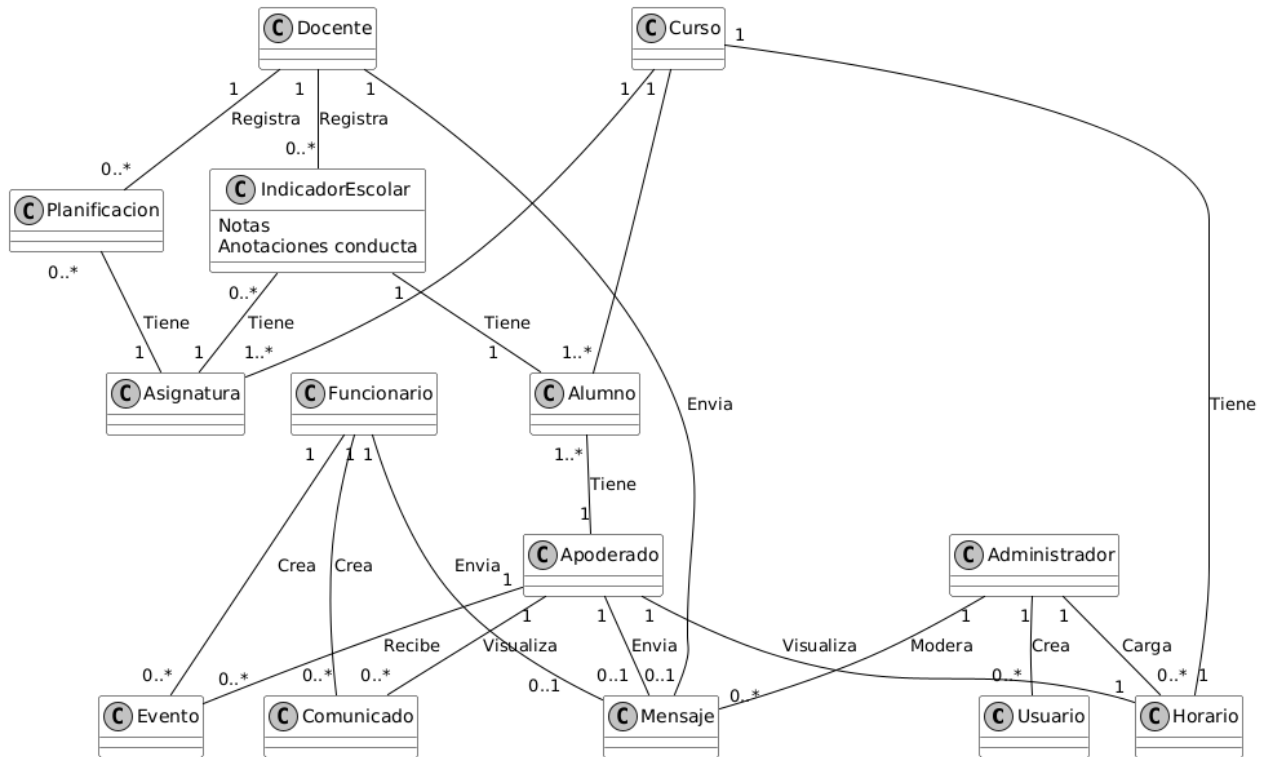


Ilustración 2-7 Modelo Conceptual
Fuente: Elaboración Propia

2.7. MODELO DE CASOS DE USO (NARRATIVO, DIAGRAMA DE SECUENCIA DEL SISTEMA Y CONTRATOS)

A continuación, se detallan los casos de uso relevantes del sistema, resultado de un análisis exhaustivo de las tareas y actividades identificadas en la estructura funcional. Cada caso de uso se presenta mediante una descripción narrativa, diagramas de secuencia y contratos, incluyendo los eventos realizados por un actor al completar un proceso.

2.7.1. CU-01 - Enviar Mensaje a apoderado

Caso de uso Narrativo:CU-01-Enviar Mensaje a apoderado

Tabla 2-21 Caso de uso Narrativo - CU-01

Caso de Uso:	Enviar Mensaje a apoderado	
Código:	CU-01	
Resumen:	El sistema debe permitir a los usuarios con rol Docente, Funcionario y Admin enviar mensajes a los usuarios con rol Apoderado.	
Tipo:	Evidente	
Ref. Cruzadas:	N/A	
Actores:	Docente / Funcionario / Admin	
Precondiciones:	El usuario debe estar registrado por un usuario administrador	
Flujo Principal		
Actor	Sistema	
1.- El actor selecciona la opción de envío de mensaje		
	2.- El sistema despliega el formulario de envío de mensaje	
3.- El actor ingresa el nombre del (los) apoderado(s) destinatarios		
	4.- El sistema despliega los nombres del(los) apoderados registrados en sistema que coincidente con el paso anterior N° 3	
5.- El actor confirma los nombres que el sistema sugiere.		
6.- El actor escribe el título del mensaje.		
7.- El actor escribe el cuerpo del mensaje.		
8.- El actor selecciona la opción de enviar el mensaje.		
	9.- El sistema valida el mensaje	
	10.- El sistema envía el mensaje.	
	11.- El sistema despliega mensaje de éxito.	
Post Condición:	El sistema envió un mensaje a un(nos) apoderados.	

Fuente: Elaboración Propia

Diagrama de Secuencia:CU-01-Enviar Mensaje a apoderado

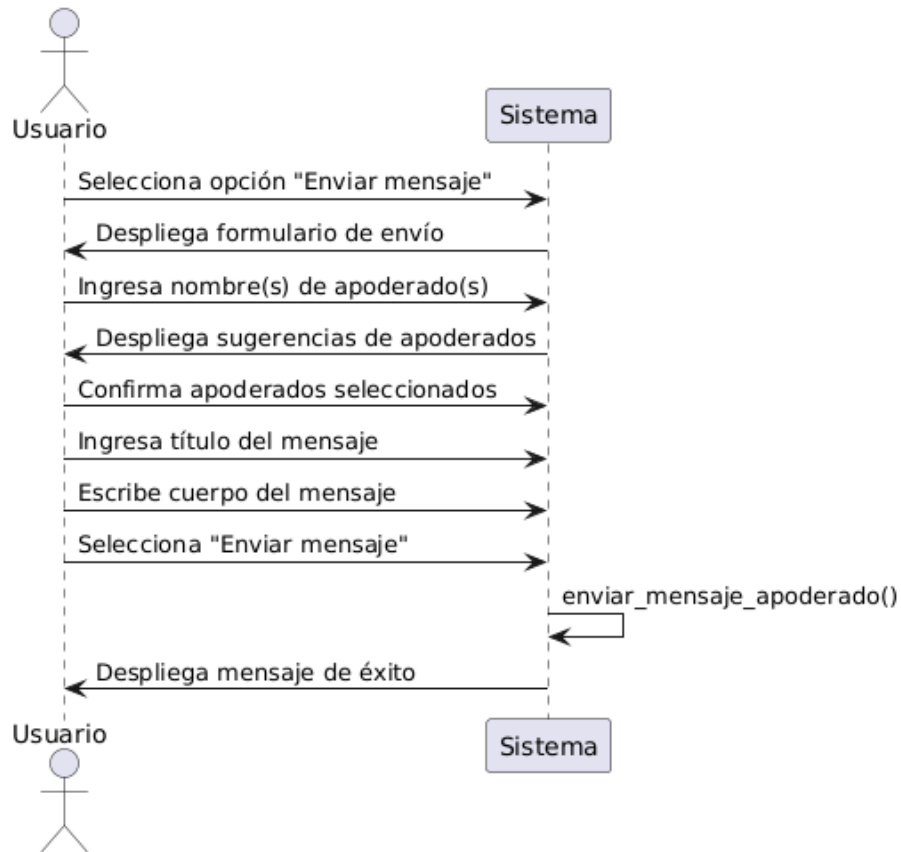


Ilustración 2-8 Diagrama de Secuencia CU-01
Fuente: Elaboración Propia

Contratos:CU-01-Enviar Mensaje a apoderado

Tabla 2-22 Contrato CU-01

Operación:	enviar_mensaje_apoderado()
Ref. Cruzadas:	CU-01- Enviar Mensaje a apoderado
Precondiciones:	<ul style="list-style-type: none"> • Usuario debe estar registrado en sistema.
Postcondiciones:	<ul style="list-style-type: none"> • Se creó una instancia de Mensaje con el título y cuerpo ingresados por el Actor. • La instancia de Mensaje se asoció a los usuarios apoderados seleccionados por el Actor. • El mensaje se envió a los apoderados seleccionados.

Fuente: Elaboración Propia

2.7.2. CU-02 - Crear evento - Entrevista apoderado

Caso de uso Narrativo: CU-02-Crear evento - Entrevista apoderado

Tabla 2-23 Caso de Uso Narrativo CU-02

Caso de Uso:	Crear evento - Entrevista apoderado	
Código:	CU-02	
Resumen:	El sistema debe permitir a los usuarios con rol Docente, Funcionario y Admin crear un <i>evento o citar a entrevista</i> a un usuario con rol apoderado	
Tipo:	Evidente	
Ref. Cruzadas:	N/A	
Actores:	Docente / Funcionario / Admin	
Precondiciones:	El usuario debe estar registrado	
Flujo Principal		
Actor	Sistema	
1.- El actor selecciona la opción "Evento - entrevista"		
	2.- El sistema despliega el formulario de envío de "Eventos - Entrevistas"	
3.- El actor ingresa el nombre del (los) apoderado(s) destinatarios		
	4.- El sistema despliega los nombres del(los) apoderados registrados en sistema que coincidente con el paso anterior N° 3	
5.- El actor confirma los nombres que el sistema sugiere.		
6.- El actor selecciona opción de Evento o Entrevista		
7.- El actor escribe descripción del evento.		
8.- El actor selecciona fecha y hora del evento.		
9.- El actor selecciona opción de confirmación y crear evento o entrevista.		
	10.- El sistema crea el evento o entrevista	
	11.- El sistema envía una notificación al usuario destinatario.	
	12.- El sistema despliega mensaje de éxito.	
Post Condición:	<ul style="list-style-type: none"> • El sistema creó un evento o entrevista • El sistema envió un mensaje de notificación al apoderado. 	

Fuente: Elaboración Propia

Diagrama de Secuencia:CU-02-Crear evento - Entrevista apoderado

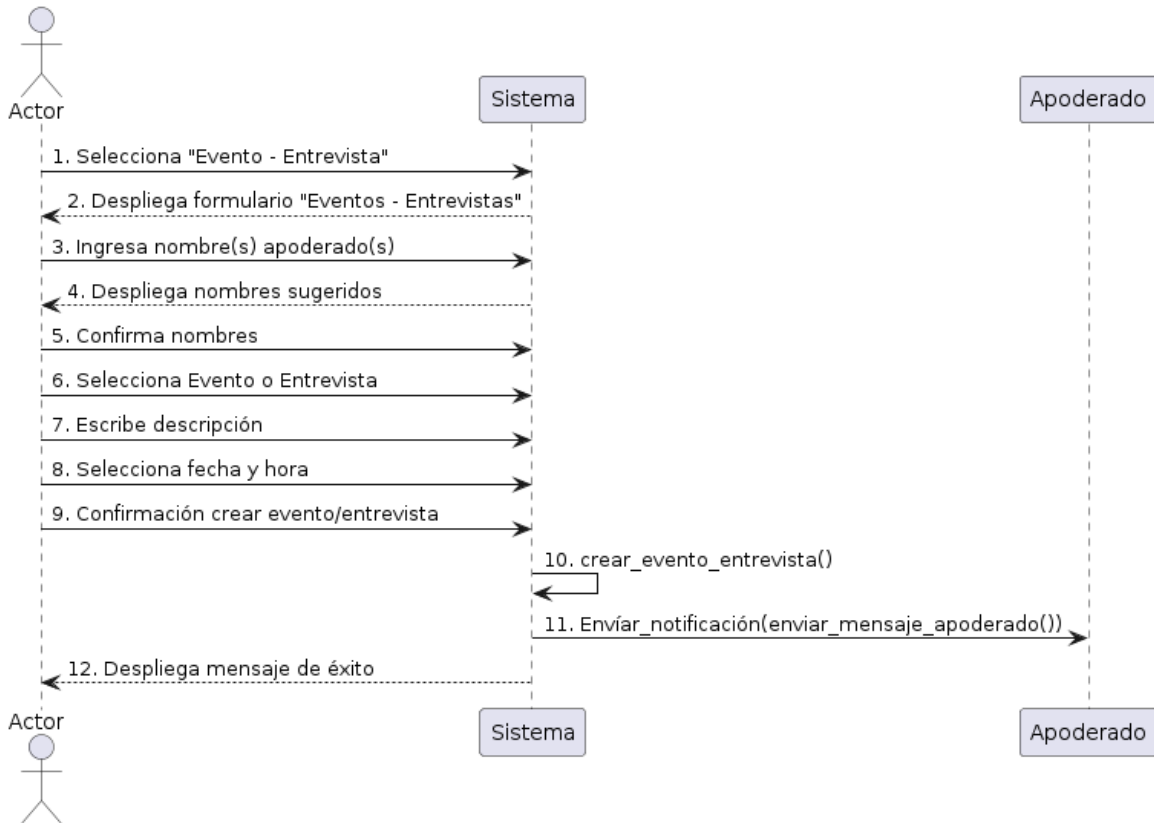


Ilustración 2-9 Diagrama de Secuencia CU-02
Fuente: Elaboración Propia

Contratos:CU-02-Crear evento - Entrevista apoderado

Tabla 2-24 Contratos CU-02

Operación:	crear_evento_entrevista()
Ref. Cruzadas:	CU-02-Crear evento - Entrevista apoderado
Precondiciones:	<ul style="list-style-type: none"> El usuario debe estar registrado en el sistema.
Postcondiciones:	<ul style="list-style-type: none"> Se creó una instancia de evento/entrevista con la descripción, fecha y hora ingresadas por el usuario. La instancia de evento/entrevista se asoció a los apoderados seleccionados por el Usuario. Se envió una notificación a los apoderados seleccionados, informándoles sobre el evento/entrevista. Se mostró un mensaje de éxito al Actor confirmando la creación del evento/entrevista.

Fuente: Elaboración Propia

2.7.3. CU-03 - Agregar - Modificar nota

Caso de uso Narrativo:CU-03 - Agregar - Modificar nota

Tabla 2-25 Caso de Uso Narrativo CU-03

Caso de Uso:	Agregar - Modificar nota	
Código:	CU-03	
Resumen:	El sistema debe permitir a los usuarios con rol Docente y Admin ingresar una nota (calificación) a un alumno registrado	
Tipo:	Evidente	
Ref. Cruzadas:	N/A	
Actores:	Docente / Admin	
Precondiciones:	El usuario debe estar registrado. Debe existir un alumno registrado con anterioridad.	
Flujo Principal		
Actor	Sistema	
1.- El actor selecciona la opción "Notas"		
	2.- El sistema despliega el formulario de "Notas"	
3.- El actor ingresa el curso del (los) alumnos.		
	4.- El sistema despliega los nombres del(los) alumnos registrados en sistema que coincidente con el paso anterior N° 3.	
5.- El actor ingresa el nombre del (los) alumnos(s).		
	6.- El sistema despliega el formulario de las notas del alumno.	
7.- El actor ingresa una nota en el formulario y/o modifica una nota existente.		
8.- El actor selecciona la opción de guardar cambios.		
	9.- El sistema valida los cambios realizados.	
	10.- El sistema realiza la transacción.	
	11.- El sistema despliega mensaje de éxito.	
Post Condición:	<ul style="list-style-type: none"> El sistema creó/modificó un registro de nota en un alumno. 	

Fuente: Elaboración Propia

Diagrama de Secuencia:CU-03 - Agregar - Modificar nota

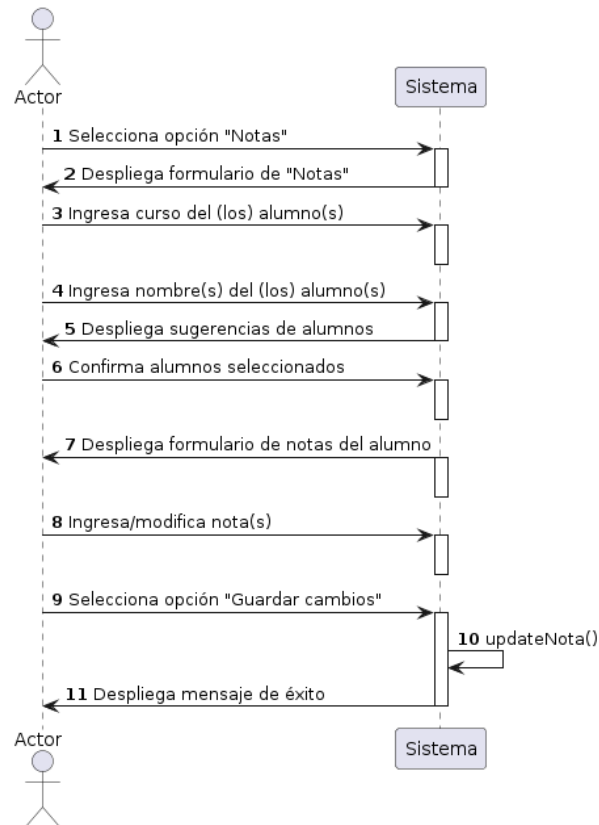


Ilustración 2-10 Diagrama de Secuencia CU-03
Fuente: Elaboración Propia

Contratos:CU-03 - Agregar - Modificar nota

Tabla 2-26 Contratos CU-03

Operación:	updateNota()
Ref. Cruzadas:	CU-03 - Agregar - Modificar nota
Precondiciones:	<ul style="list-style-type: none"> El usuario debe estar registrado en el sistema. Alumno debe existir con anterioridad.
Postcondiciones:	<ul style="list-style-type: none"> Se modificó una y/o creó una nota de un alumno en el sistema. Se mostró un mensaje de éxito al Actor confirmando la creación y/o modificación de la nota.

Fuente: Elaboración Propia

**3. CAPÍTULO 3: ASPECTOS RELEVANTES DEL DISEÑO PARA LA ALTERNATIVA
SELECCIONADA**

3.1. ARQUITECTURA DEL SOFTWARE

La arquitectura de software como indican Shaw y Garland refiere a "*la estructura general de éste y las formas en que ésta (arquitectura) entrega integridad conceptual a un sistema*" [21] considerando lo indicado anteriormente, la arquitectura de un sistema resulta un ente fundamental en el proceso de diseño de software, ya que esto permite contemplar los elementos y actividades más detallados de un sistema, permitiendo de esta forma aplicar trazabilidad y seguimiento, lo cual posibilita gestionar con mayor éxito un proyecto de software. En virtud de lo anterior se detallarán los elementos arquitectónicos de la solución seleccionada para El colegio Laico de Valdivia.

Para la solución de software seleccionada se considerarán los siguientes puntos tratados en la sección de *requerimientos no funcionales* del presente informe [22]:

- RNF-03: Concurrencia mínima de 300 usuarios
- RNF-02: Sistema debe ser multiplataforma.

En primer lugar, se debe considerar que los requerimientos no funcionales corresponden a restricciones y/o propiedades de calidad del software a desarrollar [20], por ende, estos puntos deben ser la piedra guía en la búsqueda por el safari de los patrones de diseño disponibles.

En el chequeo de las arquitecturas disponibles se tomó en cuenta la arquitectura de microservicios en función de sus siguientes propiedades según las referencias consultadas [23]:

- Diseñado para fallar (desing for failure)
- Automatización de la infraestructura.

Estas propiedades entregan una gran ventaja en caso de desastres y/o problemáticas relacionadas a infraestructura o calidad de producto (código), pudiendo en primer lugar, reemplazar y gestionar fallas usando contenedores y en caso de correcciones y/o errores estos futuros cambios pueden ser desplegados de forma oportuna.

Sin embargo, la tecnología a utilizar para la lógica de negocio (Phoenix Framework) cuenta con un patrón de arquitectura *Modelo Vista Controlador* (MVC) [24] el cual delega responsabilidades y tareas en estructuras de software que interactúan entre sí [25]. Además, la tecnología *React Native* basada en la Librería ReactJS [18], permite desarrollar software multiplataforma con una gran documentación enfocada a los microservicios, mediante el estilo de arquitectura REST.

En función de los estamentos anteriores, para la arquitectura de este proyecto se considerarán los siguientes elementos, basados en la separación de responsabilidades de acuerdo con las siguientes capas:

- *Interfaz de usuario/presentación – frontend:*
Esta capa de muestra la interfaz a los usuarios, en donde podrán realizar las operaciones que el software ofrece. Tendrá comunicación con la lógica de negocio mediante el protocolo HTTP/REST

- Lógica de negocio/ controladores – backend:
Esta capa contiene la lógica de negocio de la solución de software, la cual realiza las operaciones solicitadas por los usuarios en la capa anterior y la persistencia de datos en la base de datos. Para la comunicación con la capa frontend se usará el protocolo HTTP/REST y en el caso de la base de datos se usarán los componentes nativos que trae el framework a utilizar (ORM - Ecto). Cabe destacar que los distintos componentes del backend serán modularizados de acuerdo con su responsabilidad y la comunicación entre estos componentes será mediante HTTP/REST.

- Base de datos / storage
Esta capa corresponde a la base de datos donde se persistirán/almacenarán todos los elementos/ información que genere la solución de software. Está se comunicará exclusivamente con el backend.

En conclusión se propone para este trabajo, una arquitectura de software mixta principalmente basada en los patrones *Modelo Vista Controlador* y *Microservicios*, con un énfasis en la separación de la interfaz de los componentes del sistema. El diagrama de arquitectura (Ilustración 3-11) ilustra la arquitectura de software propuesta para el sistema de comunicaciones e informaciones escolares.

Esta arquitectura modular y escalable permite un desarrollo y mantenimiento más sencillo [23], además de facilitar la adaptación a futuros cambios y requerimientos del sistema.

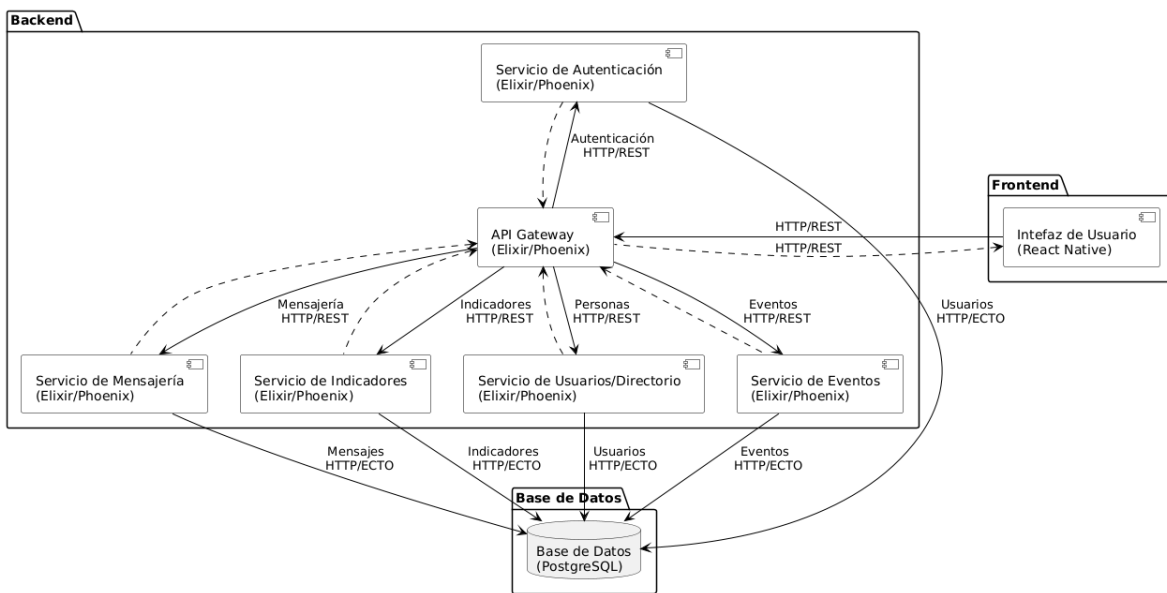


Ilustración 3-11 - Diagrama de Arquitectura
Fuente: Elaboración Propia

3.1.1. Lenguajes de programación y tecnologías

Para el desarrollo de la solución se eligió el siguiente listado de tecnologías acompañada de la correspondiente justificación de elección:

3.1.1.1. React Native

Librería cuyo génesis es el framework creado por Meta (ex Facebook) [18] ReactJS del lenguaje de programación JavaScript. Permite desarrollos multiplataforma y experiencias de usuario nativas en distintos dispositivos, ya sean móviles y web. Lo anterior permite reducir el tiempo de lanzamiento al mercado, mejorando el tiempo de desarrollo del proyecto. Contemplando lo anterior, con React Native se codificará el frontend solo una vez para los distintos dispositivos soportados: web, IOS y Android.

3.1.1.2. Elixir

Elixir es un lenguaje de programación basado en el lenguaje de programación Erlang [26], está pensado para aplicaciones que requieran alta concurrencia en tiempo real, gracias a su paradigma funcional. Lo anterior es adecuado para las especificaciones que tiene el proyecto relacionadas con la mensajería y la concurrencia de usuarios. Además, se usará el framework del lenguaje de programación elixir llamado Phoenix [24] que está diseñado para crear API's RESTful de forma robusta, aliviando al mismo el esfuerzo requerido para completar el desarrollo.

3.1.1.3. PostgreSQL

PostgreSQL (PgSQL) es una BBDD con un gran soporte gracias su larga trayectoria y exquisita documentación [27]. En el contexto de este proyecto se considera la implementación a BB.DD documental que entrega PgSQL, pudiendo entregar flexibilidad en el contexto futuro de modificaciones u iteraciones que puedan alterar y aumentar el esfuerzo necesario para el mismo menester en alguna BBDD relacional pura. También es importante indicar que PgSQL permite entregar una consistencia de datos y soportar transacciones, que son necesarias para una futura implementación del modelo de datos normado por MINEDUC [28]

3.1.2. Sistema operativo

Para el despliegue en producción de esta solución, se contempla en primera instancia usar una configuración *On Premise*, es decir que el software sea ejecutado en los servidores propios que mantiene El Colegio Laico. Debido a un proceso de migración que está realizando la institución en conjunto con la empresa MICROSOFT, se contempla además el uso de la nube pública AZURE propiedad de MICROSOFT lo anterior con un costo adicional para la institución. En consideración a lo anterior, el despliegue de este proyecto debe tomar en cuenta una instancia On Premise o Cloud bajo decisión de la institución según su capacidad financiera en el momento darse la implantación de la solución de software.

En la parte técnica se optó por el Sistema Operativo GNU/Linux en específico la distribución Debian, ofrece una base sólida para el despliegue de los servicios de la solución de software ya sea en ambiente On Premise o Cloud tales como Azure.

3.1.3. Herramientas necesarias para el desarrollo del proyecto

En virtud del software mencionado en el apartado anterior se enumerarán las herramientas necesarias para poder desarrollar la solución de software seleccionada:

3.1.3.1. Entorno de Desarrollo:

- **Sistema Operativo Debian:** Proporciona la base estable y segura sobre la cual se construirá y ejecutará todo el proyecto, además de ser el sistema operativo que se usará para el despliegue de la solución.
- **Editor de Código Visual Studio Code:** Este editor de texto ofrece un entorno de desarrollo integrado (IDE) con soporte para Elixir y JavaScript, lo que facilita la escritura, depuración y gestión del código fuente tanto del frontend como del backend.
- **Terminal:** La terminal de Debian BASH (Bourne-Again Shell) permite interactuar directamente con el sistema operativo, ejecutar comandos para instalar dependencias, iniciar servidores, gestionar la base de datos y realizar otras tareas esenciales durante el desarrollo.

- **Git:** Este sistema de control de versiones distribuido es fundamental para gestionar el historial del código, realizar seguimiento de cambios y trazabilidad del proyecto.

3.1.3.2. Backend (Elixir/Phoenix):

- **Ecto:** Esta librería de Elixir facilita la interacción con bases de datos PostgreSQL, proporcionando un conjunto de herramientas para realizar consultas, crear esquemas, gestionar transacciones y otras operaciones necesarias para la persistencia de datos.
- **Postman:** Herramienta para probar y depurar APIs. Facilitará la prueba de las APIs del backend durante el desarrollo y testeado.

3.1.3.3. Frontend (React Native):

- **Node.js y npm:** Node.js es un entorno de ejecución de JavaScript del lado del servidor y npm un gestor de paquetes, son esenciales para instalar y gestionar las dependencias de React Native, así como para ejecutar scripts de construcción y otras tareas relacionadas con el desarrollo del frontend que involucran elementos del lenguaje JavaScript.
- **Android Studio:** Este entorno de desarrollo integrado (IDE) proporciona las herramientas necesarias para desarrollar, depurar y construir aplicaciones Android.
- **Xcode:** Este IDE de Apple es esencial para desarrollar aplicaciones iOS, se usará en el proyecto para garantizar la compatibilidad y el rendimiento óptimo en dispositivos Apple.
- **Figma:** Herramienta de diseño utilizada para crear prototipos y diseños visuales de la interfaz de usuario, mejorando la comunicación entre diseñadores y desarrolladores.

3.1.3.4. Testing:

- **Jest:** Framework de pruebas de JavaScript, Se utilizará principalmente para probar los componentes de React Native.
- **Detox:** Framework de pruebas end-to-end (E2E) para aplicaciones móviles React Native. Permite simular interacciones del usuario y verificar el comportamiento de la aplicación en un entorno real.
- **ExUnit:** Framework de pruebas integrado en Elixir. Permite pruebas unitarias, de integración y de propiedad.

3.2. DISEÑO DE DATOS

En este capítulo, se exponen los distintos elementos que conforman el diseño de datos, el cual está embebido en un nivel abstracto [2], pero en el presente desarrollo se expondrán en diferentes representaciones, algunas de forma específica que definen la implementación futura del sistema a desarrollar.

3.2.1. Modelo de clases

A continuación, se presenta el modelo de clases (Ilustración 3-12) que expone la estructura de los objetos, sus relaciones y operaciones [20] en el sistema a desarrollar. Este diagrama permite visualizar las clases principales, sus atributos y métodos, así como las relaciones que existen entre ellas.

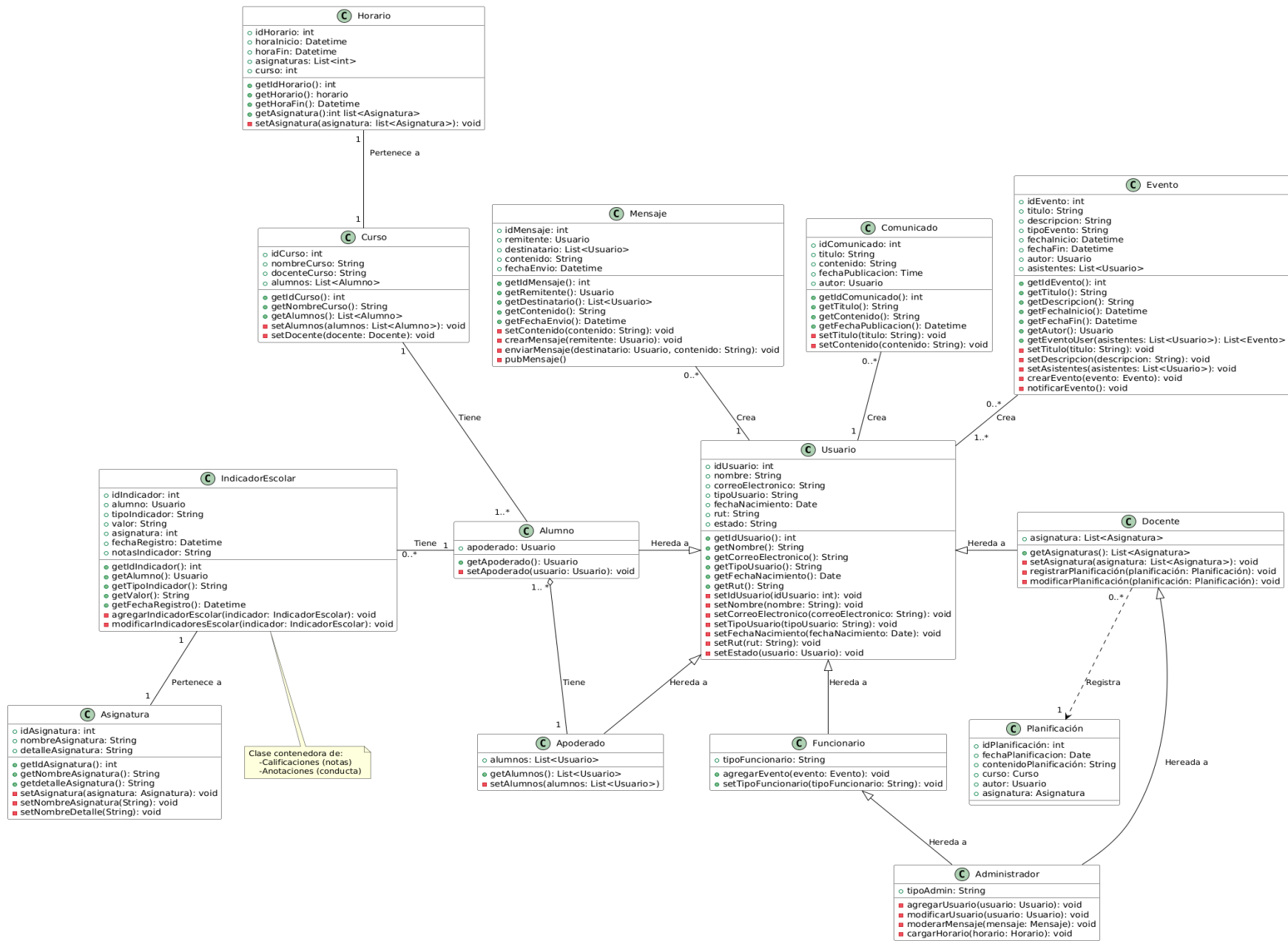


Ilustración 3-12 Diagrama de Clases
 Fuente: Elaboración Propia

3.2.2. Modelo Relacional

En un modelo relacional, los datos se almacenan en tablas con un esquema fijo, mientras que en un modelo documental como en el seleccionado para la solución (PostgreSQL), los datos se almacenan como documentos JSON, que pueden tener una estructura más flexible y variada.

Este enfoque ofrece varias ventajas, como una mayor flexibilidad para adaptar la estructura de datos a los cambios en los requisitos, un mejor rendimiento en consultas que involucran datos relacionados y una reducción en la cantidad de uniones necesarias para acceder a la información. El Diagrama de entidad relación que se presenta a continuación (Ilustración 3-13) ilustra las entidades clave, sus atributos y las relaciones entre ellas.

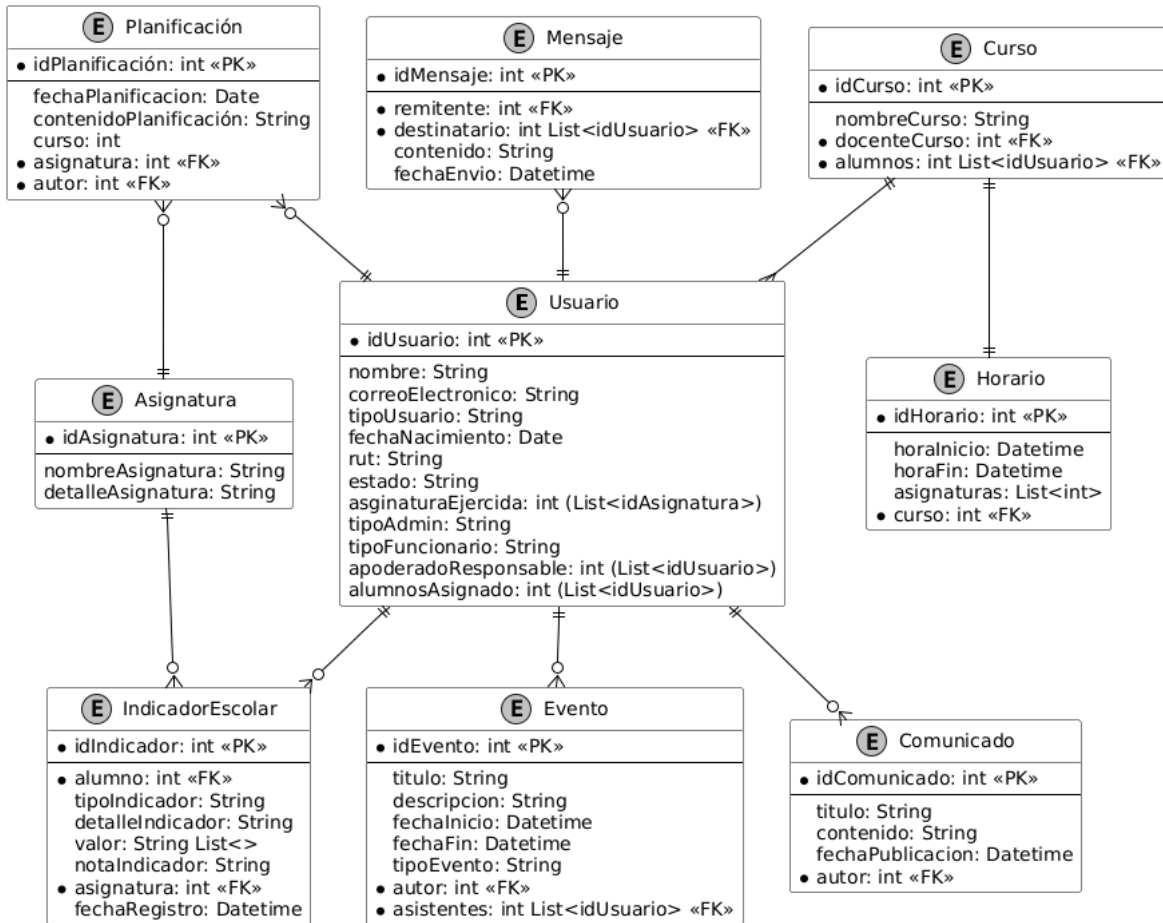


Ilustración 3-13 - Diagrama Entidad Relación
Fuente: Elaboración Propia

3.2.3. Diccionario de datos

A continuación, se presenta el diccionario de datos de la solución de software a desarrollar, con el cual se otorga consistencia y evita conflictos en el nombrado de las entidades y subentidades, además de servir como un almacén de información para el proyecto y sus futuras iteraciones [20].

3.2.3.1. Tabla Asignatura

Tabla 3-27 Tabla Asignatura

Nombre Entidad	Asignatura	
Descripción	Almacena las materias lectivas (clases) impartidas en el colegio.	
Clave Primaria	idAsignatura	
Claves Foráneas	N/A	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idAsignatura <<PK>>	int	Identificador único de la asignatura.
nombreAsignatura	String	Nombre de la asignatura.
detalleAsignatura	String	Detalle de la asignatura, que puede incluir descriptores necesarios para la práctica lectiva..

Fuente: Elaboración Propia

3.2.3.2. Tabla Comunicado

Tabla 3-28 Tabla Comunicado

Nombre Entidad	Comunicado	
Descripción	Almacena los comunicados dirigidos a la comunidad escolar, incluyendo el título, contenido y fecha de publicación.	
Clave Primaria	idComunicado	
Claves Foráneas	autor	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idComunicado <<PK>>	int	Identificador único del comunicado. Representa información dirigida a toda la comunidad escolar.
titulo	String	Título del comunicado.
contenido	String	Contenido del comunicado.
fechaPublicacion	Datetime	Fecha y hora en que se publicó el comunicado.
Autor <<FK>>	String	ID del usuario que creó el comunicado.

Fuente: Elaboración Propia

3.2.3.3. Tabla Curso

Tabla 3-29 Tabla Curso

Nombre Entidad	Alumno	
Descripción	Almacena la información de los alumnos de un curso en particular.	
Clave Primaria	idCurso	
Claves Foráneas	docenteCurso - alumnos	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idCurso <<PK>>	int	Identificador único del usuario.
nombreCurso	String	Nombre del curso.
alumnos <<FK>>	List<int>	Lista de IDs de alumnos asignados al curso.
docenteCurso <<FK>>	Int	ID del docente responsable del curso.

Elaboración Propia

3.2.3.4. Tabla Evento

Tabla 3-30 Tabla Evento

Nombre Entidad	Evento	
Descripción	Almacena los eventos creados por los usuarios.	
Clave Primaria	idEvento	
Claves Foráneas	autor - asistentes	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idEvento <<PK>>	int	Identificador único del evento. Representa actividades o fechas importantes para la comunidad escolar.
titulo	String	Título del evento.
descripcion	String	Detalle del evento creado.
fechaInicio	Datetime	Indica fecha calendarizada para el evento
fechaFin	Datetime	Fecha de finalización del evento.
tipoEvento	String	Parámetro que indica el tipo de evento generado (ENTREVISTA - EVENTO_GRAL)
autor <<FK>>	String	ID del usuario que creó el evento.
asistentes <<FK>>	List<int> >	Lista de IDs de usuarios asistentes.

Fuente: Elaboración Propia

3.2.3.5. Tabla Horario

Tabla 3-31 Tabla Horario

Nombre Entidad	Horario	
Descripción	Almacena el horario de clases de los alumnos, incluyendo el día de la semana, la hora de inicio y fin, la asignatura y el ID del alumno asociado.	
Clave Primaria	idHorario	
Claves Foráneas	curso	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idHorario <<PK>>	Int	Identificador único del horario de un alumno.
horaInicio	Datetime	Hora de inicio del horario.
horaFin	Datetime	Hora de fin del horario.
asignaturas	List<int>	Lista de IDs de asignaturas en el horario.
horaFin	Time	Hora de fin de la clase.
Curso <<FK>>	int	ID del curso asociado.

Fuente: Elaboración Propia

3.2.3.6. Tabla IndicadorEscolar

Tabla 3-32 Tabla IndicadorEscolar

Nombre Entidad	IndicadoreEscolar	
Descripción	Almacena los indicadores escolares de los alumnos, como calificaciones y notas de conducta, incluyendo el tipo de indicador, el valor, la fecha de registro y el ID del alumno asociado.	
Clave Primaria	idIndicador	
Claves Foráneas	alumno - asignatura	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idIndicador <<PK>>	int	Identificador único del indicador escolar.
alumno <<FK>>	int	Identificador del alumno al que pertenece el indicador.
tipoIndicador	String	Tipo de indicador (CALIFICACION, ANOTACION_CONDUCTA).
notasIndicador	String	Descripción detallada del indicador, campo requerido cuando se ingresa un indicador con valor tipoIndicador ANOTACIÓN_CONDUCTA .
valor	List<String>	Valor del indicador (nota, calificación, etc.).
fechaRegistro	Datetime	Fecha y hora en que se registró el indicador.
Asignatura <<FK>>	int	ID de la asignatura relacionada.

Fuente: Elaboración Propia

3.2.3.7. Tabla Mensaje

Tabla 3-33 Tabla Mensaje

Nombre Entidad	Mensaje	
Descripción	Almacena los mensajes enviados entre usuarios del sistema, incluyendo el contenido, la fecha de envío y los IDs del remitente y destinatario.	
Clave Primaria	idMensaje	
Claves Foráneas	remitente - destinatario	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idMensaje <<PK>>	int	Identificador único del mensaje.
Remitente <<FK>>	int	Identificador del usuario que envió el mensaje.
Destinatario <<FK>>	List<int>	Lista de IDs de usuarios destinatarios.
Contenido	String	Contenido del mensaje.
fechaEnvio	Datetime	Fecha y hora en que se envió el mensaje.

Fuente: Elaboración Propia

3.2.3.8. Tabla Planificación

Tabla 3-34 Tabla Mensaje

Nombre Entidad	Planificación	
Descripción	Almacena el registro de la actividad lectiva ocurrida en una clase de alguna asignatura, impartida por un docente a un curso.	
Clave Primaria	idPlanificación	
Claves Foráneas	asignatura - autor	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idPlanificación <<PK>>	int	Identificador único de la planificación.
fechaPlanificacion	Datetime	Fecha de la planificación.
contenidoPlanificacion	String	Contenido de la planificación.
curso	String	ID del curso relacionado.
asignatura <<FK>>	int	ID de la asignatura relacionada.
autor<<FK>>	int	ID del usuario con rol DOCENTE que creó la planificación.

Fuente: Elaboración Propia

3.2.3.9. Tabla Usuario

Tabla 3-35 Tabla Usuario

Nombre Entidad	Usuario	
Descripción	Almacena la información de los usuarios del sistema, incluyendo administradores, apoderados, docentes y funcionarios. Cada registro representa un usuario único con sus datos personales y rol en el sistema.	
Clave Primaria	idUsuario	
Claves Foráneas	N/A	
Estructura del Registro		
Atributo	Tipo de Dato	Descripción Atributo
idUsuario	int	Identificador único del usuario.
nombre	String	Nombre completo del usuario.
correoElectronico	String	Correo electrónico del usuario.
tipoUsuario	String	Tipo de usuario, describe el rol en el sistema (ADMINISTRADOR, APODERADO, DOCENTE, FUNCIONARIO, ALUMNO).
fechaNacimiento	Date	Fecha de nacimiento del usuario.
rut	String	RUT del usuario
estado	String	Estado actual del usuario (Activo, Desactivo, Suspendido).
asignaturaEjercida	List<int>	Lista de id de asignaturas ejercidas por el usuario en caso de que tenga el rol Docente.
tipoAdmin	String	Describe el cargo de un usuario con rol ASMINISTRADOR (informático - soporte)
tipoFuncionario	String	Describe el cargo de un usuario con rol FUNCIONARIO (Inspector - psicólogo - Paradocente - Administrativo)
apoderadoResponsable	List<int>	Lista de IDs de apoderados responsables, en caso de que el usuario tenga el rol ALUMNO
alumnosAsignados	List<int>	Lista de IDs de alumnos asignados, en caso de que el usuario tenga el rol APODERADO

Elaboración Propia

3.3. DIAGRAMAS DE SECUENCIA EXTENDIDO Y DE COLABORACIÓN

Los diagramas de secuencia en esencia son usados para mostrar las comunicaciones entre los objetos que componen un sistema durante la ejecución de una tarea [2]. En este apartado se detallarán los diagramas de secuencia extendidos de las tareas principales, los cuales presentan bucles, condiciones y secuencias alternativas, con la finalidad de entregar una mejor comprensión del comportamiento del sistema.

3.3.1. Diagrama de Secuencia CU-01 – Enviar Mensaje a Apoderado

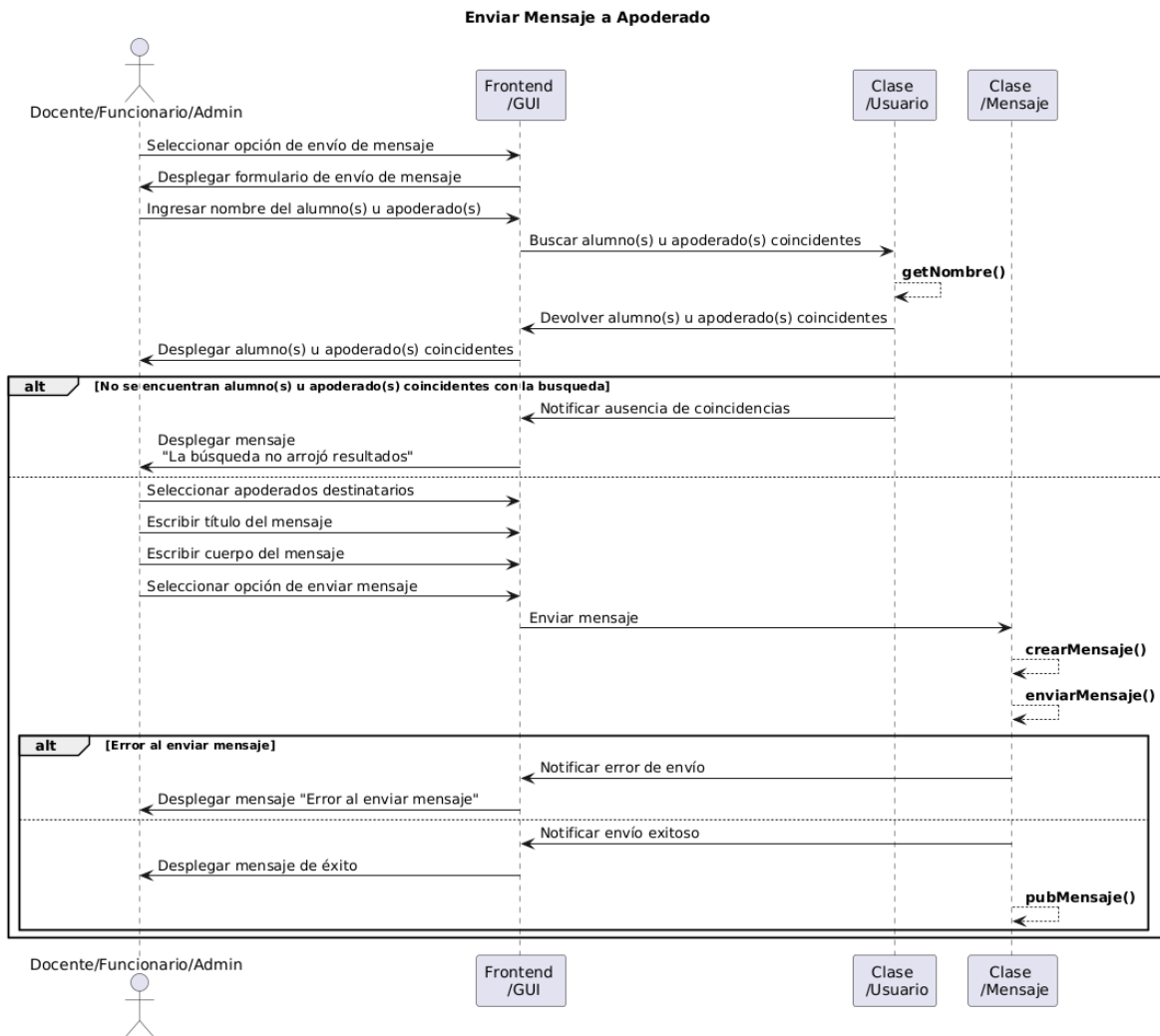


Ilustración 3-14 - Diagrama de secuencia Extendido CU-01.
Fuente: Elaboración Propia.

3.3.2. Diagrama de Secuencia CU-02 – Crear Evento / Entrevista a Apoderado

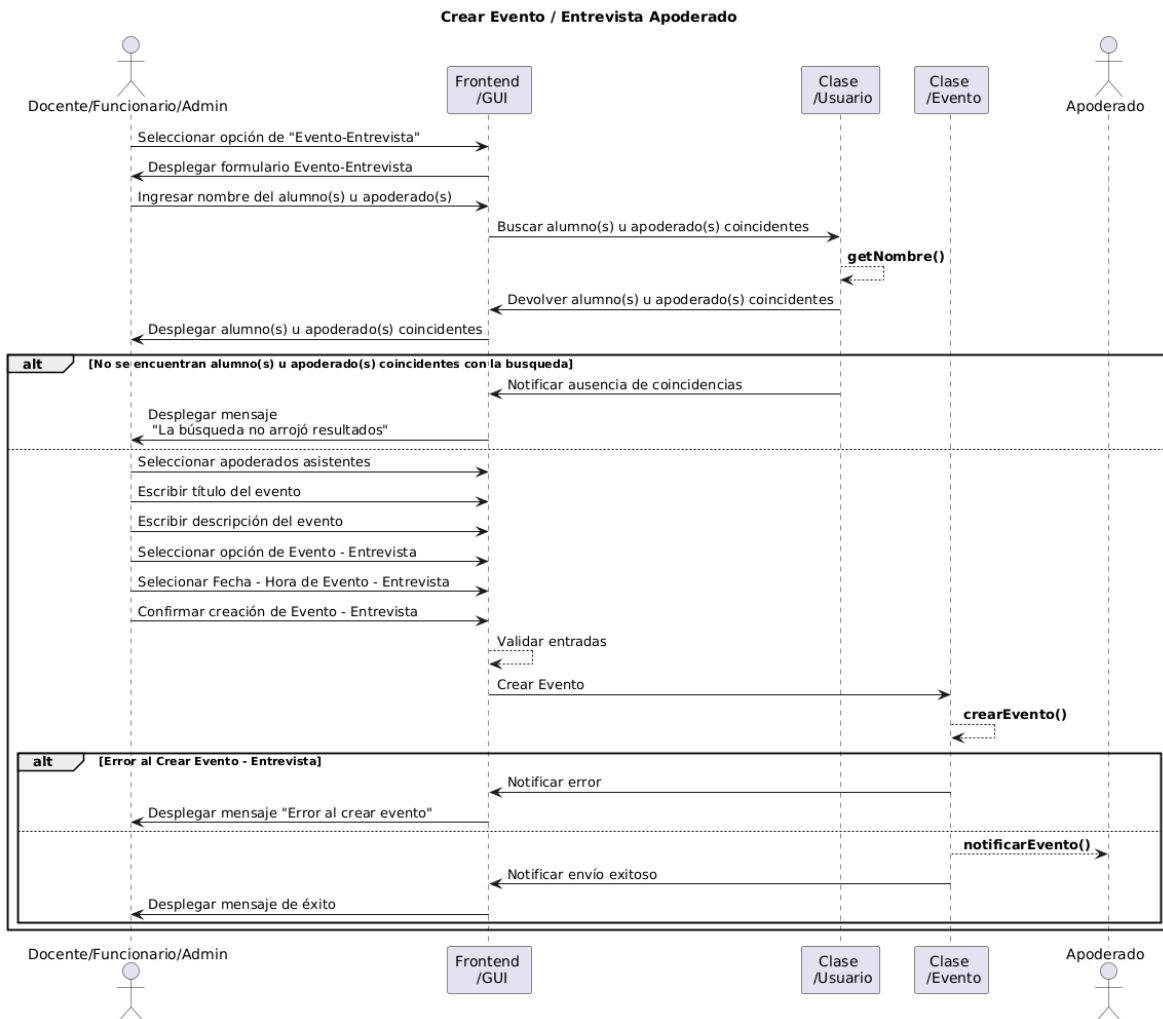


Ilustración 3-15 Diagrama de secuencia Extendido CU-02.
Fuente: Elaboración Propia.

3.3.3. Diagrama de Secuencia CU-03 – Agregar / Modificar nota

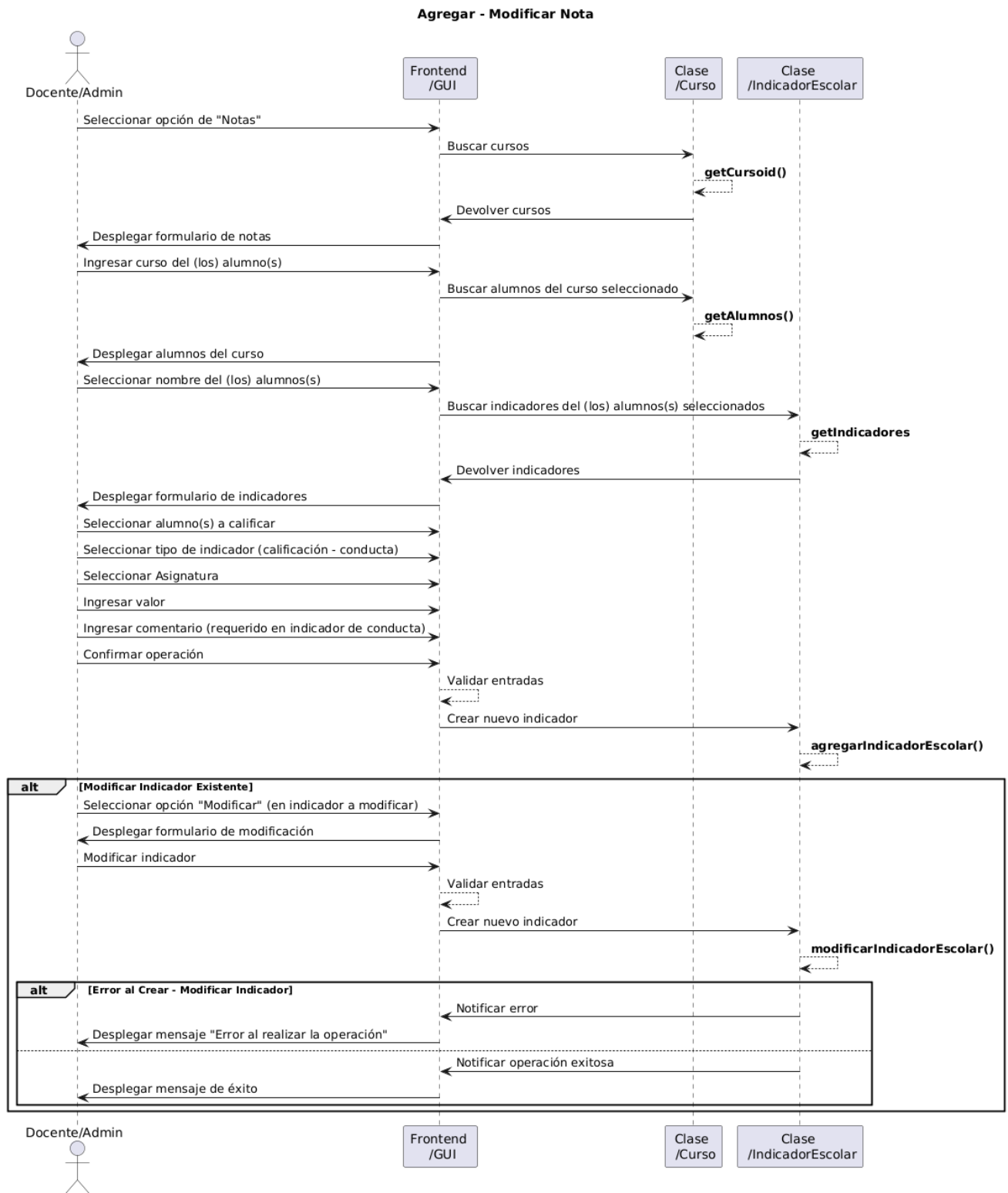


Ilustración 3-16 Diagrama de secuencia Extendido CU-03.
Fuente: Elaboración Propia.

3.4. DISEÑO DE INTERFAZ

En la actualidad las interfaces de usuario tienen una importancia crucial esto en parte, al vivir rodeados de productos de alta tecnología que requieren interacción humana. [2] La eficiencia y éxito de un producto de software, está principalmente radicada en un ajuste entre las habilidades, experiencia y expectativa de los usuarios previstos [20].

3.4.1. Ingeniería de Interfaces y experiencia de usuario

Para la creación de prototipos e interfaces que ofrezcan a los usuarios una experiencia eficiente y cómoda, el diseño de esta solución de software seguirá los lineamientos de Nielsen [29]. Estos lineamientos presentan 10 heurísticas que proporcionan principios claros para desarrollar sistemas funcionales, accesibles y fáciles de usar, reduciendo errores y maximizando la satisfacción del usuario.

A continuación, se describen las 10 heurísticas de Nielsen y su aplicación prioritaria en los módulos principales, así como en otros elementos gráficos y funcionales que conforman la solución de mensajería (Tabla 3-35Tabla).

Tabla 3-36 Aplicación de heurísticas de Nielsen

Heurística de Nielsen	Aplicación en los Diseños
1. Visibilidad del estado del sistema	<ul style="list-style-type: none"> - En Mensajería, los indicadores de mensajes no leídos y confirmaciones de lectura mantienen al usuario informado. - En Comunicados, se usan alertas visuales para mostrar comunicados no leídos.
2. Correspondencia entre el sistema y el mundo	<ul style="list-style-type: none"> - En Eventos, se utilizan tarjetas para mostrar los eventos y entrevistas recibidas. - Los módulos usan términos y categorías comprensibles ("Conducta" para anotaciones de comportamiento). - Home mantiene un símil a un tablero de anuncios, mostrando un resumen con la información reciente del usuario. - En Horario, se mantiene un símil con el horario de clases distribuido en papel.
3. Control y libertad del usuario	<ul style="list-style-type: none"> - En Mensajería, los usuarios pueden editar mensajes, ya enviados.
4. Consistencia y estándares	<ul style="list-style-type: none"> - Interfaz uniforme en todos los módulos (colores, tipografía, y diseño). - Íconos comunes en todos los módulos (como el de calendario, gráficos y notificaciones).
5. Prevención de errores	<ul style="list-style-type: none"> - En los formularios donde se ingresen datos, se incluyen descripciones (placeholder) de los datos solicitados en cada entrada de texto.
6. Reconocimiento en lugar de recuerdo	<ul style="list-style-type: none"> - Uso de listas y vistas predefinidas (chats recientes primero en Mensajería). -Íconos y colores intuitivos para identificar categorías rápidamente (azul para positivos, rojo para negativos en Conducta, iconos de acuerdo con la acción).
7. Flexibilidad y eficiencia de uso	<ul style="list-style-type: none"> - En Mensajería y Home, la lista de mensaje y otros elementos refleja gran parte de la mayoría de información de cada ítem (Fecha, remitente, etc.)
8. Estética y diseño minimalista	<ul style="list-style-type: none"> - Las interfaces tienen jerarquía visual clara: solo elementos esenciales en cada pantalla. - Uso de espacio en blanco para evitar sobrecarga cognitiva, especialmente en Comunicados e Indicadores.
9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores	<ul style="list-style-type: none"> - Mensajes de error claros y específicos en formularios ("El correo ingresado no es válido", etc.) - Indicadores visuales de problemas en Mensajería (ícono de mensaje no enviado debido a problemas de conexión).
10. Ayuda y documentación	<ul style="list-style-type: none"> - En cada módulo, un botón o sección de ayuda contextual con tutoriales breves. -Descripciones breves de cada función o campo editable.

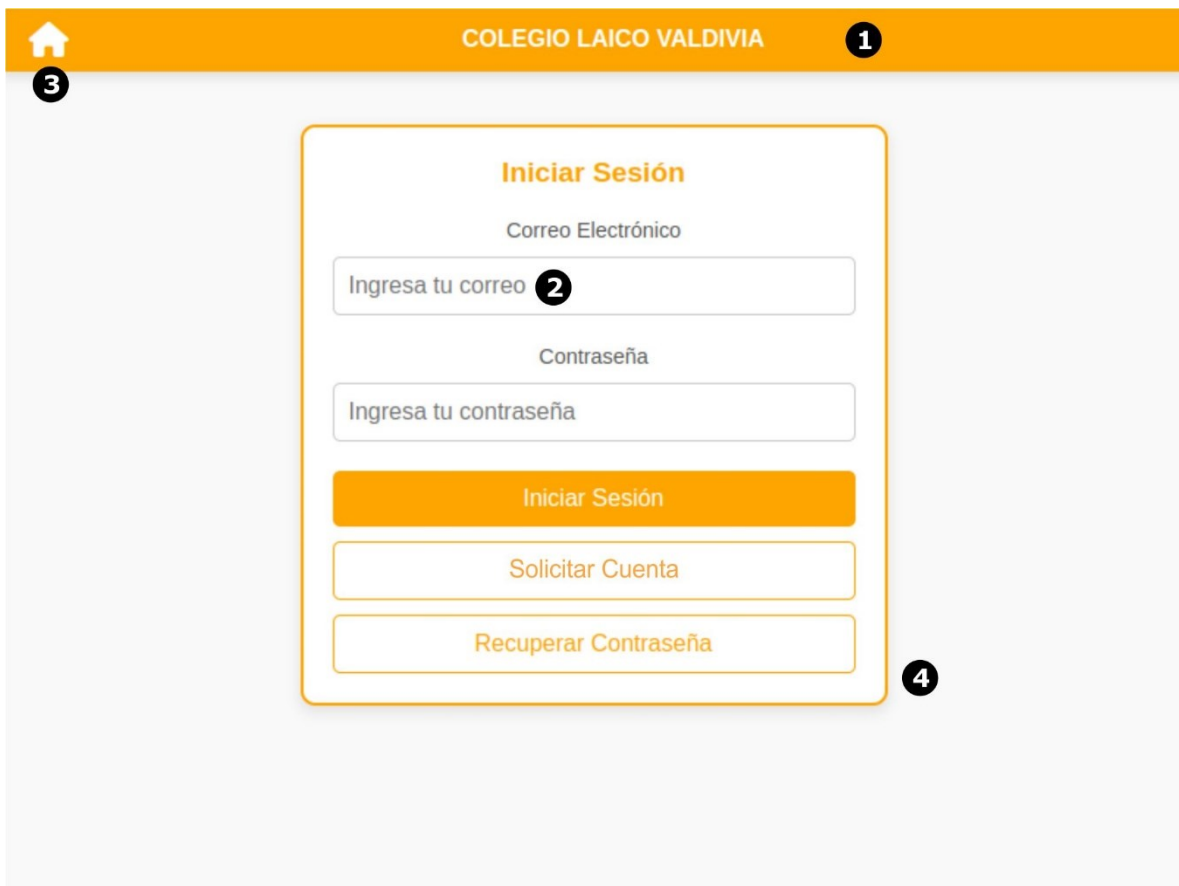
Fuente: Usability Engineering [29] / Elaboración Propia

3.4.2. Prototipo de interfaz

A continuación, se presentan prototipos de las páginas más importantes de la solución de software a desarrollar en versión de escritorio y móvil, que ilustran la funcionalidad básica y el flujo de usuario previsto. Estos esquemas permiten visualizar la interacción entre las diferentes secciones y elementos del sitio, facilitando la identificación de posibles mejoras y ajustes.

En cada prototipo se documenta las decisiones de diseño contempladas en función de la usabilidad.

3.4.2.1. Login



El prototipo muestra una interfaz de usuario para el login de escritorio. En la parte superior, hay una barra de navegación naranja con un icono de inicio (3) a la izquierda, el texto "COLEGIO LAICO VALDIVIA" en el centro y un círculo con el número "1" a la derecha. El contenido principal es un formulario con el título "Iniciar Sesión". Debajo del título, se indica "Correo Electrónico" y hay un campo de texto con el placeholder "Ingresa tu correo" (2). A continuación, se indica "Contraseña" y hay un campo de texto con el placeholder "Ingresa tu contraseña". Debajo de estos campos, hay tres botones: "Iniciar Sesión" (4), "Solicitar Cuenta" y "Recuperar Contraseña".

Ilustración 3-17 Prototipo Login Desktop
Fuente: Elaboración Propia

Simbología: Aplicación de heurísticas de Nielsen en Prototipo Login

- 1- Consistencia y estándares:** Se mantendrá el mismo diseño del header y la barra de navegación en todos los elementos gráficos de la solución, garantizando uso consistente;
- 2- Prevención de errores:** Todos los campos que requieran entrada de datos incluirán un placeholder que describa la información necesaria, reduciendo la probabilidad de errores.
- 3- Reconocimiento en lugar de recuerdo:** Los encabezados y barras de navegación del sistema incluirán iconos intuitivos, identificando categorías y acciones disponibles.
- 4- Estética y diseño minimalista:** Se usarán espacios en blanco y colores institucionales claros, para evitar la sobrecarga cognitiva. Además, se mantendrá una jerarquía visual clara, mostrando únicamente los elementos esenciales en cada pantalla.

El prototipo muestra una pantalla de inicio de sesión con un encabezado naranja que contiene un ícono de casa y el texto "COLEGIO LAICO VALDIVIA". El contenido principal está dentro de un recuadro con borde naranja y contiene:

- El título "Iniciar Sesión" en naranja.
- El encabezado "Correo Electrónico" en gris.
- Un campo de entrada con el placeholder "Ingresa tu correo".
- El encabezado "Contraseña" en gris.
- Un campo de entrada con el placeholder "Ingresa tu contraseña".
- Un botón naranja con el texto "Iniciar Sesión".
- Un botón con borde naranja y texto "Solicitar Cuenta".
- Un botón con borde naranja y texto "Recuperar Contraseña".

Ilustración 3-18 Prototipo Login Móvil
Fuente: Elaboración Propia

3.4.2.2. Home

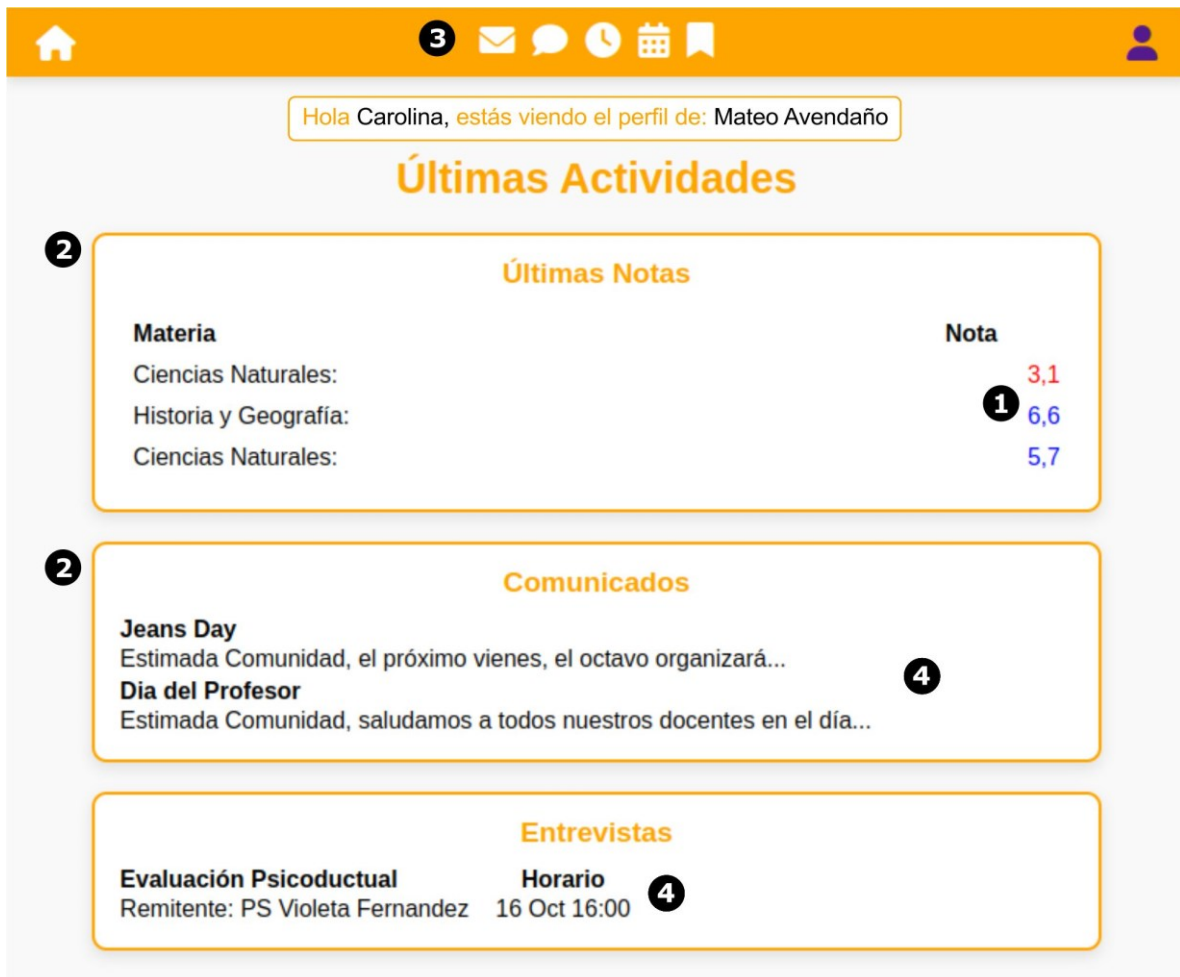


Ilustración 3-19 Prototipo Home Desktop
Fuente: Elaboración Propia

Simbología: Aplicación de heurísticas de Nielsen en Prototipo Home

- 1- Reconocimiento en lugar de recuerdo:** Se utilizarán colores diferenciados para representar las distintas notas.
- 2- Correspondencia entre el sistema y el mundo real:** El diseño del home se presenta como un símil de un tablero, presentando la información más reciente del usuario.
- 3- Reconocimiento en lugar de recuerdo:** Los encabezados y barras de navegación del sistema incluirán iconos intuitivos, identificando categorías y acciones disponibles.
- 4- Flexibilidad y eficiencia de uso:** En el home se mostrará la mayor cantidad posible de información relevante sobre cada elemento. Con la finalidad de evitar abrir cada elemento de manera individual.

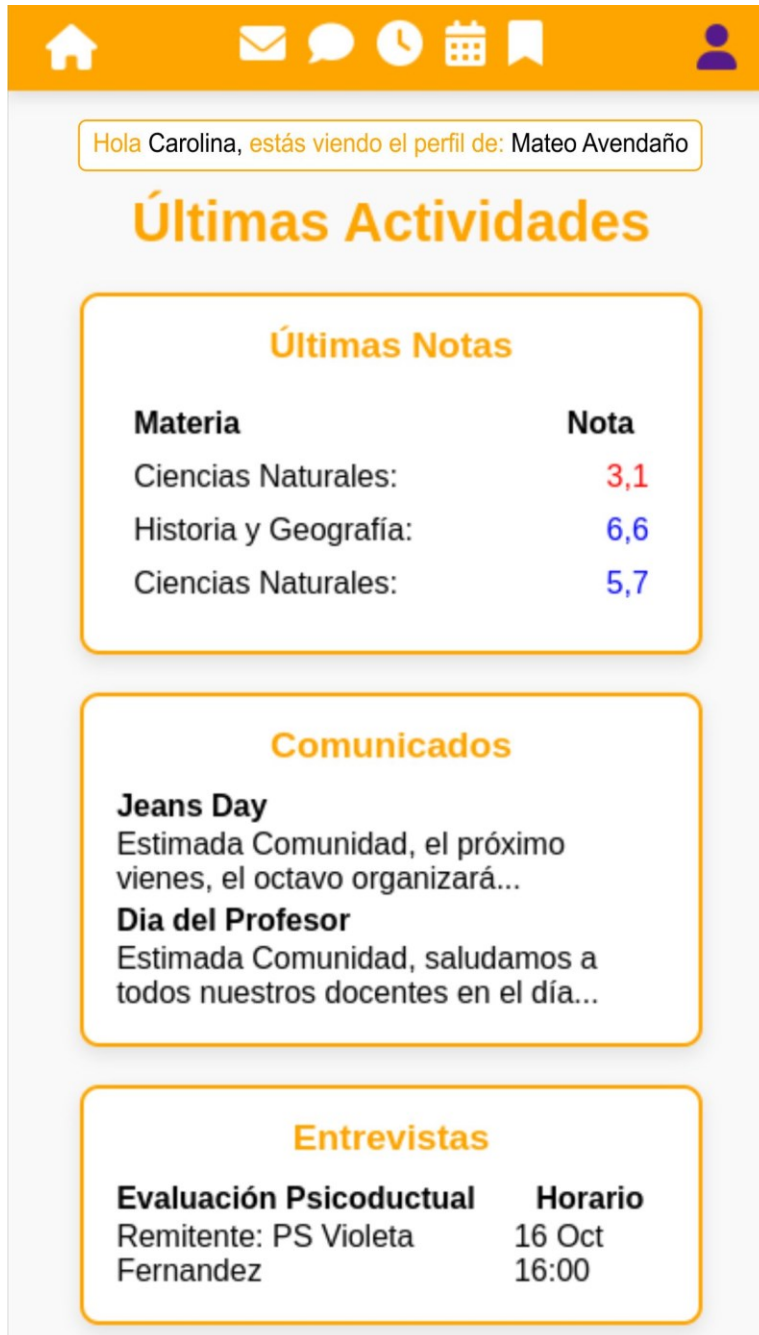


Ilustración 3-20 Prototipo Home Móvil
Fuente: Elaboración Propia

3.4.2.3. Mensajería

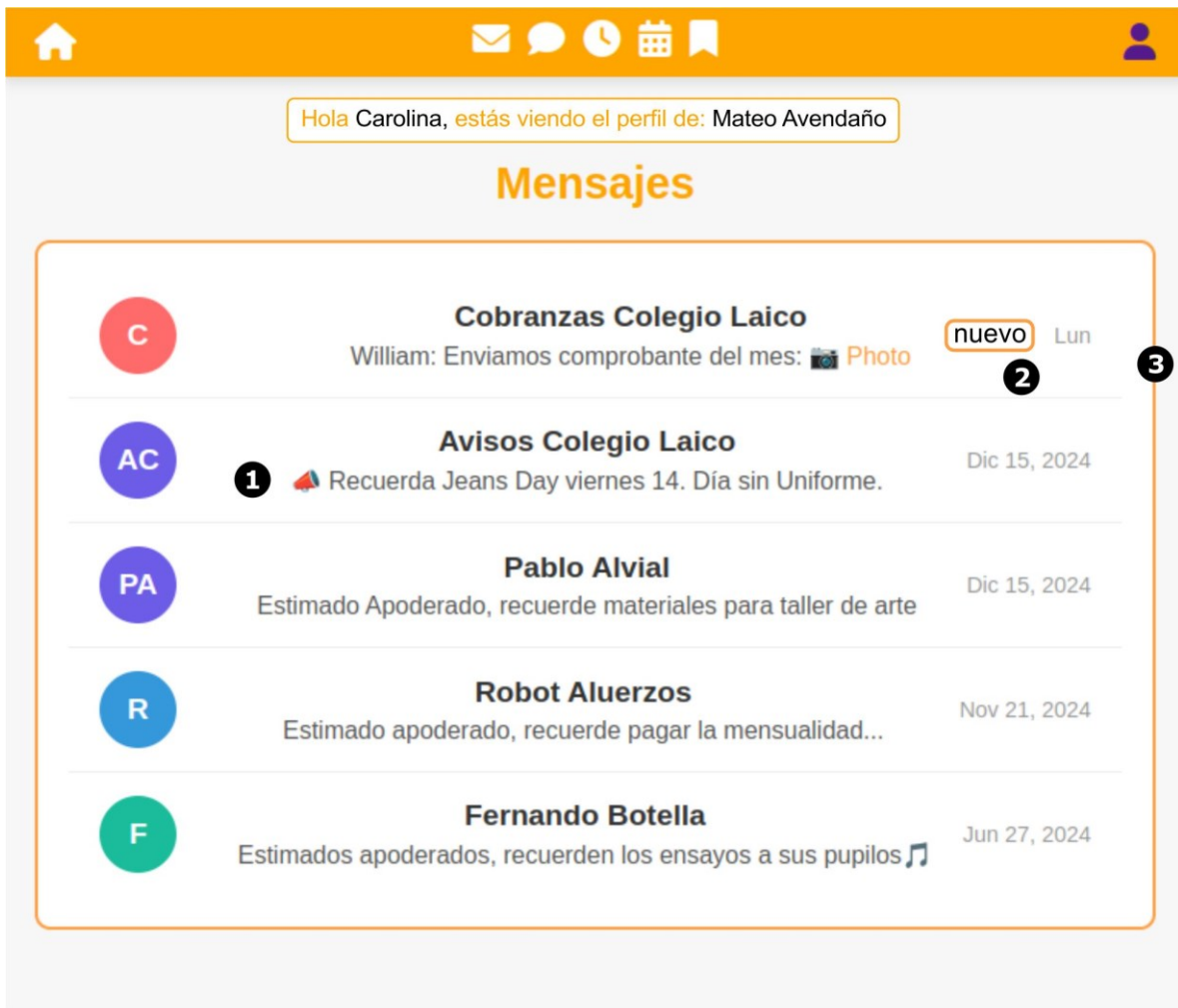


Ilustración 3-21 Prototipo Mensajería Desktop
Fuente: Elaboración Propia

Simbología: Aplicación de heurísticas de Nielsen en Prototipo Mensajería

- 1- Flexibilidad y eficiencia de uso:** En cada mensaje en esta pantalla se mostrará la mayor cantidad posible de información de acuerdo con la resolución de pantalla del usuario.
- 2- Visibilidad del estado del sistema:** En cada mensaje no leído se posiciona un indicador para dar visibilidad al estado de lectura del mensaje por parte del usuario.
- 3- Correspondencia entre el sistema y el mundo real:** Se ordenarán los mensajes de forma cronológica, desde los más recientes a los más antiguos.



Ilustración 3-22 Prototipo Mensajería Móvil
Fuente: Elaboración Propia

3.4.2.4. Horarios

Horario
4° Básico B

Horas	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-8:45	Lenguaje Camila F.	Lenguaje Camila F.	Taller Intercult.	Inglés Y. Contreras	C. Naturales Camila F.
8:45-9:30	Lenguaje Camila F.	Lenguaje Camila F.	Taller Intercult.	Inglés Y. Contreras	C. Naturales Camila F.
9:30-9:45	Recreo				
9:45-10:30	Matemáticas Camila F.	Historia Camila F.	Taller Hab. Luis T.	E. Física O. Alcayaga	Lenguaje Camila F.
10:30-11:15	Matemáticas Camila F.	Historia Camila F.	Taller Hab. Luis T.	E. Física O. Alcayaga	Lenguaje Camila F.
11:15-11:30	Recreo				
11:30-12:15	Historia Camila F.	E. Física O. Alcayaga	Orientación Camila F.	Artes Camila F.	Matemáticas C. Farías
12:15-1:00	Almuerzo				
1:00-1:45	C. Naturales Camila F.	E. Física O. Alcayaga	Lenguaje Camila F.	Artes Camila F.	
1:45-2:30	Taller Socioem. Sebastián O.	Música Camila F.	Tecnología J. Aracena	Física y Religión Sebastián O.	
2:30-3:15	Taller Socioem. Sebastián O.	Música Camila F.	Tecnología J. Aracena	Física y Religión Sebastián O.	

Ilustración 3-23 Prototipo Horario Desktop
Fuente: Elaboración Propia

Simbología: Aplicación de heurísticas de Nielsen en Prototipo Horario

- 1- Correspondencia entre el sistema y el mundo real:** Se presentará el horario de clases como un símil al horario usado y distribuido actualmente en papel.



Horario

4° Básico B

Horas	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-8:45	Lenguaje Camila F.	Lenguaje Camila F.	Taller Intercult.	Inglés Y. Contreras	C. Naturales Camila F.
8:45-9:30	Lenguaje Camila F.	Lenguaje Camila F.	Taller Intercult.	Inglés Y. Contreras	C. Naturales Camila F.
9:30-9:45	Recreo				
9:45-10:30	Matemáticas Camila F.	Historia Camila F.	Taller Hab. Luis T.	E. Física O. Alcayaga	Lenguaje Camila F.
10:30-11:15	Matemáticas Camila F.	Historia Camila F.	Taller Hab. Luis T.	E. Física O. Alcayaga	Lenguaje Camila F.
11:15-11:30	Recreo				
11:30-12:15	Historia Camila F.	E. Física O. Alcayaga	Orientación Camila F.	Artes Camila F.	Matemáticas C. Farías
12:15-				

Ilustración 3-24 Prototipo Horario Móvil (vista horizontal)
Fuente: Elaboración Propia

3.4.2.5. Eventos



Ilustración 3-25 Prototipo Eventos Desktop
Fuente: Elaboración Propia

Simbología: Aplicación de heurísticas de Nielsen en Prototipo Mensajería

- 1- Visibilidad del estado del sistema:** En cada evento no leído se posiciona un indicador para dar visibilidad al estado de lectura del evento por parte del usuario.
- 2- Estética y diseño minimalista:** Cada evento estará contenido en una tarjeta de forma individual con fondo blanco, con la finalidad de evitar la sobrecarga visual.
- 3- Correspondencia entre el sistema y el mundo real:** Se ordenarán los eventos y entrevista de forma cronológica, desde los más próximos a los más remotos.



Ilustración 3-26 Prototipo Eventos Móvil
Fuente: Elaboración Propia

3.4.3. Diagrama de menús

La Tabla muestra la jerarquía de navegación de la solución de software a desarrollar y las secciones y funcionalidades que componen cada página. El orden de navegación se representa donde el menor valor es el primer elemento de la jerarquía.

Tabla 3-37 Diagrama de Paginas

Orden	Nombre Pagina	Secciones - Descripción
0	Login	Se realiza Ingreso al sistema y permite recuperar contraseña
1	Home	Bienvenida y resumen del contenido de la página
		Indicadores escolares de cada alumno asignado, como asistencia, calificaciones, y comportamiento
2	Mensajería	Interfaz principal para ver y gestionar los mensajes recibidos y emitidos
		Función de búsqueda con lupa para buscar en el directorio de personas
		Formulario para redactar y enviar un nuevo mensaje
2	Indicadores Escolares	Título de la página y breve descripción de los indicadores escolares
		Explicación de la funcionalidad de la grilla de indicadores escolares
		Presentación visual de algunos indicadores escolares clave en gráficos
		Grilla de indicadores escolares basada en los registros de la base de datos
2	Comunicados	Contenido detallado de los comunicados, incluyendo fechas, detalles y cualquier acción requerida.
		Lista de comunicados anteriores y archivados para referencia futura.
2	Eventos	Encabezado que introduce la lista de eventos destacados.
		Lista de eventos próximos y pasados con detalles como fecha, hora y descripción.
2	Solicitud de entrevista	Título y breve descripción sobre la solicitud de entrevista
		Formulario para que los usuarios puedan solicitar una entrevista, incluyendo campos como nombre, correo electrónico, fecha y hora preferida, y motivo de la entrevista
2	Directorio	Lista de categorías de contactos (Administradores, Profesores, Funcionarios)
		Detalles de cada contacto, incluyendo nombre, posición, correo electrónico y teléfono
		Formulario de contacto para enviar mensajes a los miembros del directorio
2	Admin - Mantenedor	Encabezado con título "Admin - Mantenedor" y subtítulo "Sección exclusiva para administradores"
		Gestión de usuarios: Crear, editar y eliminar usuarios
		Gestión de roles: Asignar y modificar roles de usuario
		Configuración del sistema: Ajustes generales y de seguridad

Fuente: Elaboración Propia

CONCLUSIONES – RECOMENDACIONES

En el presente trabajo, se ha detallado el análisis de ingeniería y posterior diseño, necesario para una posible implementación de un sistema de mensajería y asuntos escolares para el Colegio Laico de Valdivia, mediante un enfoque sistemático obtenido con rigor durante tres años en los placidos atardeceres en la sede UTFSM San Joaquín.

En primer lugar se realiza la génesis identificando una problemática cercana y dispuesta a solventar con alguna solución de software, la cual es puesta al agudo escrutinio ingenieril, donde se reversan y contonean los primeros pasos de una posible solución: ¿Qué necesita el usuario? ¿Cuánto costará? ¿Cómo lo realizaremos?, entre otros. Esto nos entrega la primera parte con una materia prima preparada para el siguiente paso.

En este segundo paso es necesario definir las acciones de los futuros usuarios de este software, quienes serán en primera instancia, aquellos que alabarán o condenarán la calidad de esta solución. Es menester pensar y contemplar paso a paso de un usuario en esta solución y quizá encuerarse en estos futuros usuarios, al fin y al cabo, todos somos usuarios.

En el último paso, se requiere una vista de ingeniero más fina, ya que toca una tarea compleja: *definir una estructura modular y escalable, que interactúa de manera eficiente para cubrir los casos de uso prioritarios*. Hay que agradecer a la prole de ingenieros y sabios del software que nos han entregado herramientas, tales como patrones de diseño, modularidad, reusabilidad, entre otros. Gracias a aquello es posible seleccionar tecnologías que proporcionan seguridad y escalabilidad en caso de alguna proyección en la cantidad de usuarios.

Para concluir cabe indicar que la concepción de la solución de software contenida en este trabajo está pensada para colegios pequeños, tal como el Colegio Laico de Valdivia, los cuales reciben fondos estatales y pueden pagar desarrollo a medida, por lo cual el desarrollo ingenieril de este trabajo da la posibilidad de exportar a otras entidades educativas con similares características y necesidades. Este enfoque abre la posibilidad de replicar esta solución en otras entidades generando en un futuro próximo una oportunidad para el autor.

Por último es grato terminar con estas palabras en especial el agradecimiento por el aprendizaje de todas las técnicas de ingeniería, que hacen posible elevar cualquier idea, necesidad y quizá hasta locuras en productos software concretos. Se agradece profundamente al cuerpo docente vespertino y a todos quienes han contribuido a este viaje de aprendizaje y creación.

BIBLIOGRAFÍA

- [1] Colegio Laico Valdivia, «Reglamento Interno Colegio Laico,» 2021. [En línea]. Available: <https://cdnsae.mineduc.cl/documentos/22459/ReglamentodeConvivencia22459.pdf>. [Último acceso: 15 2 2024].
- [2] R. S. Pressman, Ingeniería de Software, Un Enfoque Práctico, 2005.
- [3] «Colegio Laico Resumen - Admisión Escolar,» MINEDUC, 2024. [En línea]. Available: <https://admission.mineduc.cl/vitrina-vue/establecimiento/22459>. [Último acceso: 15 2 2024].
- [4] Colegio Laico Valdivia, «Proyecto Educativo - Colegio Laico Valdivia,» 2000. [En línea]. Available: <https://colegiolaicovaldivia.cl/wp-content/uploads/2022/05/ProyectoEducativo-2022.pdf>. [Último acceso: 15 2 2024].
- [5] Bibliote del Congreso Nacional de Chile, «Ley 20845,» 2024. [En línea]. Available: Ley 20845. [Último acceso: 15 2 2024].
- [6] Gran Logia de Chile. , «Masoneria Public Home | Web Publica,» 2023. [En línea]. Available: https://www.granlogia.cl/nuestros_principios. [Último acceso: 10 Noviembre 2024].
- [7] L. Ceci, «Chat apps and messaging dau,» Statista, 16 2 2022. [En línea]. Available: <https://www.statista.com/statistics/1365686/dau-communication-apps-worldwide>. [Último acceso: 15 2 2024].
- [8] Papinotas, «Papinotas - Plataforma de Gestión Escolar,» Papinotas. [En línea]. [Último acceso: 15 2 2024].
- [9] KIMCHE, «El libro de clases digital más amigable y seguro,» KIMCHE, 12 4 2024. [En línea]. Available: <https://www.kimche.co/>. [Último acceso: 15 2 2025].
- [10] S. e. a. Caro-Álvaro, «Applying usability recommendations when developing mobile instant messaging applications,» *IET Software*, vol. 16(1), nº doi:10.1049/sfw2.12039. , p. 73–93, 2021.
- [11] Papinotas, «Así funciona Papinotas,» 22 12 2013. [En línea]. Available: <https://www.youtube.com/watch?v=5GprlruWQcA>. [Último acceso: 5 5 2024].
- [12] Biblioteca del Congreso Nacional, «Ley 21096 - CONSAGRA EL DERECHO A PROTECCIÓN DE LOS DATOS PERSONALES,» 16 6 2018. [En línea]. Available: <https://www.bcn.cl/leychile/navegar?idNorma=1119730>. [Último acceso: 15 2 2024].
- [13] BCN, «Ley 19628 - SOBRE PROTECCION DE LA VIDA PRIVADA,» 28 8 1999. [En línea]. Available: <https://www.bcn.cl/leychile/navegar?idNorma=141599>. [Último acceso: 15 2 2024].
- [14] ENTEL, «Las redes sociales en Chile 2023 más usadas y cuántos usuarios tienen,» 15 2 2023. [En línea]. Available: <https://ce.entel.cl/articulos/las-redes-sociales-en-chile/>. [Último acceso: 15 2 2024].
- [15] KImche, «Comunicaciones - ZOHODESK,» 5 Mayo 2024. [En línea]. Available: <https://kimche.zohodesk.com/portal/es/kb/kimche-comunicaciones>. [Último acceso: 5 Mayo 2024].
- [16] Project Management Institute, Inc, PMBOOK, 2017.
- [17] xmpp.org, «XMPP | An Overview of XMPP,» [En línea]. Available: <https://xmpp.org/about/technology-overview/>. [Último acceso: 10 Noviembre 2024].
- [18] Meta, «<https://reactnative.dev/>,» [En línea]. Available: <https://reactnative.dev/>.

- [19] talent.com, [En línea]. Available: <https://cl.talent.com/>.
- [20] I. A. a. E. A. Sommerville, Ingeniería del software, Madrid: Pearson Educación, 2005.
- [21] M. y. D. G. Shaw, «Formulations and Formalisms in Software Architecture,» 1995.
- [22] C. Fica, «ANÁLISIS E IMPLEMENTACIÓN DE SOLUCIÓN DE MENSAJERÍA Y ASUNTOS ESCOLARES PARA EL COLEGIO LAICO DE VALDIVIA,» USM, Viña, 2024.
- [23] J. L. a. M. Fowler, ""Microservices," martinowler.com," 25 Mar 2014. [Online]. Available: <https://martinowler.com/articles/microservices.html>.
- [24] DockYard, «phoenix framework,» [En línea]. Available: <https://www.phoenixframework.org/>.
- [25] M. Fowler, Patterns of Enterprise Application Architecture, 2002.
- [26] elixir team, «Elixir Lang,» [En línea]. Available: <https://elixir-lang.org/>.
- [27] The PostgreSQL Global Development Group, «PostgreSQL: The world's most advanced open source database,» [En línea]. Available: <https://www.postgresql.org/>.
- [28] mineduc, «Estándar de datos para la Educación,» [En línea]. Available: <https://www.ede.mineduc.cl/>.
- [29] J. Nielsen, Usability inspection methods. In Conference companion on Human factors in, ACM, 1994.
- [30] IEEE, «IEEE Recommended Practice for Software Requirements Specifications,» 20 Oct. 1998. [En línea]. Available: doi: 10.1109/IEEESTD.1998.88286. [Último acceso: 1 Jun 2024].
- [31] J. Hertvik, "Service Availability: Calculations and Metrics, Five 9s, and Best Practices," BMC Blogs, [Online]. Available: <https://www.bmc.com/blogs/service-availability-calculation-metrics/>.

GLOSARIO – ANEXOS

API's RESTful: Conjunto de reglas y principios para diseñar servicios web que se comunican a través de HTTP.

Azure: Plataforma de computación en la nube de Microsoft que ofrece una amplia gama de servicios, como almacenamiento, procesamiento y redes.

Backend: Parte de una aplicación que se ejecuta en el servidor y se encarga de la lógica de negocio, el acceso a datos y otras operaciones internas.

BASH: Shell de Unix y lenguaje de scripting ampliamente utilizado en sistemas GNU/Linux.

BB.DD (Bases de Datos): Colección organizada de datos estructurados que se almacenan y gestionan de manera eficiente.

Detox: Framework para pruebas end-to-end de aplicaciones móviles React Native.

ECTO: Biblioteca de Elixir para trabajar con bases de datos relacionales.

End-to-End: Tipo de prueba que verifica el funcionamiento completo de una aplicación, desde la interfaz de usuario hasta el backend.

Erlang: Lenguaje de programación concurrente diseñado para construir sistemas altamente escalables y tolerantes a fallos.

ExUnit: Framework de pruebas unitarias para el lenguaje Elixir.

Figma: Herramienta de diseño colaborativo basada en la web.

FOSS: Acrónimo de software libre de código Abierto (Free Open Source Software)

Framework: Estructura de software que facilita el desarrollo de aplicaciones, proporcionando herramientas y patrones comunes.

GNU/Linux: Familia de Sistemas operativos de código abierto basado en el kernel Linux y GNU utils.

Meta (antes Facebook, Inc.): Empresa multinacional de tecnología. Anteriormente conocida como Facebook, Inc., cambió su nombre en 2021 para reflejar su enfoque en el metaverso.

MICROSOFT: Empresa multinacional de tecnología que desarrolla, fabrica, licencia y vende software, hardware y servicios.

On Premise: Modelo de implementación de software en el que la aplicación se instala y ejecuta en los servidores de la propia organización.

Phoenix: Framework web para el lenguaje de programación elixir que permite construir aplicaciones web de alto rendimiento.

PostgreSQL: Sistema de gestión de bases de datos relacional de código abierto y altamente escalable.

POSTMAN: Herramienta para probar y desarrollar API's RESTful.

ReactJS: Biblioteca de JavaScript para construir interfaces de usuario interactivas y dinámicas.

Wireframes: Representación visual básica de la estructura y el diseño de una interfaz de usuario.

Xcode: Entorno de desarrollo integrado (IDE) de Apple para crear aplicaciones para macOS, iOS, iPadOS, watchOS y tvOS.