



UNIVERSIDAD TECNOLÓGICA METROPOLITANA
FACULTAD DE INGENIERÍA
ESCUELA DE INFORMÁTICA

AppSISAV, Proyecto App móvil para usuarios de la plataforma
SISAV

TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN MENCIÓN INFORMÁTICA

AUTOR:

ROA INOSTROZA, NELSON HERNAN

PROFESOR GUÍA:

CASTILLO VALDEZ, MAURO

SANTIAGO – CHILE

2020

Autorización para la Reproducción del Trabajo de Titulación

1. Identificación del trabajo de titulación

Nombre del(os) alumno(s): NELSON HERNAN ROA INOSTROZA

Rut: 18.367.582-6

Dirección: PASAJE PISCIS 01928, PUENTE ALTO

E-mail: NELSON.ROAI@UTEM.CL

Teléfono: +56961123325

Título de la tesis: AppSISAV, Proyecto App móvil para usuarios de la plataforma SISAV

Escuela: INFORMÁTICA

Carrera o programa: INGENIERÍA CIVIL EN COMPUTACION MENCION INFORMATICA

Título al que opta: TRABAJO DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN MENCIÓN INFORMÁTICA

2. Autorización de Reproducción (seleccione una opción)

- a) Este trabajo de titulación no puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso escrito del(os) autor(es), exceptuando la cita bibliográfica, resumen y metadatos que acreditan al trabajo y a su(s) autor(es).

Fecha: _____ Firma: _____

- b) Se autoriza la reproducción total o parcial de este trabajo de titulación, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica que acredita al trabajo y a su autor.

En consideración a lo anterior, se autoriza su reproducción de forma (marque con una X):

<input type="checkbox"/>	Inmediata
<input type="checkbox"/>	A partir de la siguiente fecha: _____ (mes/año)

Fecha: _____ Firma: _____

Esta autorización se otorga en el marco de la ley N°17.336 sobre Propiedad Intelectual, con carácter gratuito y no exclusivo para la Institución.

NOTA OBTENIDA:_____

Firma y timbre autoridad
responsable

DEDICATORIA

Dedico este trabajo de título principalmente a mis padres, Patricia y Nelson, que fueron mis más grandes apoyos durante todo mi proceso formativo, quienes me apoyaron siempre en todo sentido, dándome confianza y tranquilidad ante las adversidades. A Katherine, mi compañera durante todos estos años, mi más grande apoyo emocional y a quien le debo en gran parte mi presencia en esta etapa de mi vida, sin ella seguramente no habría llegado a esta instancia. A mis compañeros de universidad, quienes fueron compañeros de trabajos y proyectos durante todo este periodo, con quienes compartí la mayor parte de estos años y sirvieron de ayuda en momentos de crisis.

AGRADECIMIENTOS

Los agradecimientos van enfocados principalmente a la casa de estudio, por darme las herramientas para poder desempeñarme de buena forma en el mercado laboral actual, agradecimientos a plataformas como UDEMY, que fueron de gran ayuda con cursos complementarios, a todos aquellos que indirectamente fueron parte de este proceso y agradecer con mucho afecto a todos aquellos que con sus comentarios desafortunados aportaron en aumentar el grado de estrés, ansiedad y frustración en esta etapa tan importante, ya que enseñan y educan a no cometer los mismos errores, a contener y consolar a los que luchan por educarse.

TABLA DE CONTENIDOS

Autorización para la Reproducción del Trabajo de Titulación	2
DEDICATORIA	5
AGRADECIMIENTOS	6
TABLA DE CONTENIDOS	7
ÍNDICE DE TABLAS	12
ÍNDICE DE ILUSTRACIONES	13
RESUMEN	14
PALABRAS CLAVES	15
ABSTRACT	16
KEYWORDS	17
1.1 Descripción del proyecto	19
1.2 Objetivos	20
1.2.1 Objetivo general	20
1.2.2 Objetivos específicos	20
1.3 Alcances y limitaciones	21
1.3.1 Alcances	21
1.3.2 Limitaciones	21
1.4 Metodología	22
1.4.1 Cómo funciona	24
1.4.2 Actividades del proyecto: Scrum	25
1.4.2.1 Fase 0- Product Backlog	27
1.4.2.2 Fase 1 - Sprint Backlog	27
1.4.2.3 Fase 2 - Sprint	27
1.4.2.4 Fase 3 - Producto	28

1.5 ANÁLISIS Y RESULTADOS	28
1.5.1 Diagnóstico	28
1.5.2 Sistema SISAV actual	31
1.5.2.1 Login	31
1.5.2.2 Acceso principal (Perfil visita)	32
1.5.2.3 Opciones menú general	32
1.5.3 Oportunidades de Mejora	34
1.5.4 Beneficios cuantitativos y cualitativos al implementar App SISAV	35
2.1 Descripción de la organización	37
2.1.1 Historia	37
2.1.2 Vinculación con el medio	38
2.1.3 Vinculación con el medio: UTEM	38
2.1.4 Estructura Organizacional	39
2.1.5 Permisos según niveles de usuario	39
2.1.6 Organigrama de la organización	40
2.1.7 Usuarios de la plataforma	41
2.2 Descripción del problema en estudio	42
2.3 Requerimientos	42
2.3.1 Hardware	43
2.3.2 Software	43
2.3.3 Humanware	43
3.1 Descripciones y evaluación de las alternativas	44
3.1.1 Descripción	44
3.1.2 Ponderación peso según alternativas	45
3.1.3 Ponderaciones de todas las alternativas según criterios de evaluación	
48	

3.2 Beneficios de la solución	50
4.1 Requerimientos de Información	52
4.2 Requerimientos Técnicos, de Control y Funcionales	52
4.3 ESTUDIO DE FACTIBILIDAD	53
4.3.1 Factibilidad Técnica	53
4.3.2 Económica	54
4.3.3 Operativa	54
5.1 DISEÑO LÓGICO	55
5.1.1 Diagramas	55
5.1.1.1 Diagrama de flujo de datos	55
5.1.1.2 Diagrama de contexto	56
5.1.2 Diseño de interfaces y descripción	56
5.2 Diseño físico	57
5.2.1 Modelamiento	57
5.2.1.1 Modelo Entidad Relación	57
5.2.1.2 Modelo Relacional	58
5.2.2 Análisis y Evaluación de Riesgos	59
5.3 Vistas	61
5.3.1 Login	61
5.3.2 Acceso principal	61
5.3.3 Menú de opciones (IPHONE)	62
5.3.4 Menú de opciones (tipo galaxy S5)	64
5.3.5 Vista lista programas	65
5.3.6 Vista lista proyectos	66
6.2 App móvil	67
6.3 Bases de datos	68

6.4 Frontend	70
6.4.1 Desarrollo frontend: Frameworks y Lenguajes	71
6.4.1.1 HTML + CSS + JavaScript	71
6.4 Backend	72
7.1 Introducción	73
7.2 Sistema de bases de datos	74
7.3 Back-end (API REST)	75
7.3.1 Controladores (Controllers)	75
7.3.2 Modelos (Models)	75
7.3.3 Config (config.json)	76
7.3.4 Rutas (Routes)	77
7.3.5 App (app.js)	77
7.3.6 www.js	78
7.4 FRONTEND	78
7.4.1 Imágenes (Carpeta imágenes)	78
7.4.2 SRC	79
7.4.3 Theme	79
7.4.4 Environments	79
7.4.5 Assets	79
7.4.6 App	80
7.4.7 Components	80
7.4.8 Interfaces	80
7.4.9 Services	81
7.4.10 Pages	81
8.1 Sistema informático	81
8.1.1 Clasificación de sistemas informáticos	83

8.1.1.1 Sistemas de apoyo a la toma de decisiones	83
8.1.1.2 Sistema de control de procesos de negocio	83
8.1.1.3 Sistemas de colaboración empresarial	84
8.1.1.4 Sistemas de Información Ejecutiva	84
8.1.1.5 Sistemas de procesamiento de transacciones	85
8.1.1.6 Sistemas de Información de Gestión	85
6.4.1.2 Frameworks de Javascript	86
6.4.1 Lenguajes BACKEND	91

RESUMEN

Esta propuesta de trabajo de titulación plantea el desarrollo de una App móvil para el “Sistema de Seguimiento de Actividades de Vinculación con el Medio” (SISAV) de la Universidad Tecnológica Metropolitana (UTEM), con el fin de mejorar y facilitar la experiencia de los usuarios de la plataforma.

El proyecto está pensado como versión inicial, abarcando las funciones más importantes de la versión web, y agregando mejoras en estas mismas, apuntando a facilitar el ingreso a los usuarios desde cualquier dispositivo móvil. Se define como reducida ya que se plantea que en primera instancia solo se permiten alguna de las funciones principales de la versión actual, dando pie para que en un futuro se vayan agregando las demás funciones, apuntando entonces a ser un símil de la versión web, incluso mejorada al optimizar procesos.

Dentro de las funciones más destacadas que tendrá la versión móvil están:

Evaluación digital por parte del comité evaluador: Se permitirá descargar documentos subidos por el solicitante, verificar si cumple con los requisitos y evaluar según este estado.

Firma digital: El comité evaluador al aprobar la solicitud (todos deberán aprobar), automáticamente validará una firma digital que se ubicará en el final de la hoja, quedando disponible para imprimir por parte del solicitante.

El solicitante podrá imprimir el documento firmado sin necesidad de salir a buscar firmas, ya que tendrá desde la aplicación la posibilidad de imprimir el documento validado y firmado.

El proyecto se desarrollará dentro del margen de la metodología ágil, siendo su desarrollo guiado por el modelo Espiral, usando también tecnologías como Ionic, para el desarrollo de aplicaciones móviles, servidor web y utilizando las tecnologías que actualmente se usan en la versión web de la página, tales como el motor de base de datos, con el fin de facilitar la implementación.

En un comienzo se analizará el funcionamiento actual del sistema de SISAV, para observar los niveles de usuario, conocer la experiencia actual del usuario promedio con el fin de plantear mejoras y facilitar los procesos ya existentes. Después de esto se desarrollará la aplicación y se implementará, mejorando la experiencia actual de los usuarios de la plataforma SISAV.

PALABRAS CLAVES

Validación electrónica, Vinculación con el medio, SISAV, IONIC, APP, Scrum, NodeJS, TypeScript, AngularJS, MySQL, Express, Bizagui.

ABSTRACT

This proposal of work of title raises the development of a mobile App for the "System of Monitoring of Activities of Linkage with the Environment" (SISAV) of the UTEM, in order to improve and facilitate the user experience of the platform.

The project is designed as an initial version, covering the most important functions of the web version, and adding improvements in these, aiming to facilitate the entry to users from any mobile device. It is defined as reduced since it is stated that in the first instance only some of the main functions of the current version are allowed, giving rise to the fact that in the future the other functions will be added, aiming then to be a simile of the web version, even improved when optimizing processes.

Among the most important functions that the mobile version will have are:

- Digital evaluation by the evaluation committee: It will be possible to download documents uploaded by the applicant, verify if it meets the requirements and evaluate according to this state.
- Digital signature: The evaluation committee when approving the application (all must approve), will automatically validate a digital signature that will be located at the end of the page, being available for printing by the applicant.

The applicant can print the signed document without having to go out to look for signatures, since it will have the possibility of printing the validated and signed document from the application.

The project will be developed within the scope of the agile methodology, being its development guided by the Spiral Model methodology, also using technologies such as Ionic, for the development of mobile applications, web server and using the technologies currently used in the web version of the page, such as the database engine, in order to facilitate implementation.

Initially, the current operation of the SISAV system will be analyzed, to observe user levels, to know the current experience of the average user in order to propose improvements and facilitate existing processes. After this the application will be developed and implemented, improving the current experience of the users of the SISAV platform.

KEYWORDS

Link with the environment, electronic validation, SISAV, IONIC, APP, Scrum, NodeJS, TypeScript, AngularJS, MySQL, Express, Bizagi.

Capítulo 1: Introducción

La comunicación es algo esencial para el ser humano, debido a las interacciones que debe someterse en el día a día. Existen muchas formas de comunicarse, tales como la comunicación escrita, física u oral, de las cuales se utilizan para poder transmitir mensajes de cualquier tipo, estas se consumen por medio de la visión (escrita y física) y audición (oral).

Si bien pensamos en el teléfono móvil como algo relativamente nuevo, podemos concluir que lo es si nos enfocamos en los orígenes de las comunicaciones, sin embargo fue en el año 1984 cuando se comenzó a comercializar el primer teléfono inalámbrico, y fue 11 años antes, es decir en 1973, cuando se dio a conocer el primer prototipo.

Pero luego de esto surge la interrogante, ¿en qué momento el teléfono celular se convierte en algo accesible para el usuario promedio? ya que en un principio era impensado que las personas de estatus económico promedio pudieran acceder a estos enormes aparatos de comunicación. Como respuesta podemos señalar que a fines de los 90 se populariza, y es solo a fines de los 90 cuando ya toma un tamaño y precio acorde a las necesidades.

El celular nos brinda la posibilidad de movernos comunicados, privacidad en las comunicaciones, instrumento de auxilio y de geolocalización, por ende seguridad personal y es una fuente inagotable de servicios.

Hoy en día no solo es eso, sino que aporta entretenimiento, redes sociales, trabajo, videoconferencias, solicitar comida, aplicaciones capaces de brindarnos cualquier tipo de servicio.

El Sistema de Seguimiento de Actividades de Vinculación con el Medio es el nombre del sistema actual que se encarga de dar seguimiento y postular a nuevas actividades, junto con analizar los montos a prestar para estos proyectos, analizarlos y aprobarlos o rechazarlos según se justifique. Actualmente cuenta con funciones de escritorio y carece de una aplicación para poder, desde la comodidad del teléfono móvil realizar funciones mínimas de aprobación y análisis de proyectos, y acceso a los medios propios de un móvil tales como carga de archivos desde la cámara del móvil y subida de documentación propia de este.

Es entonces cuando surge la pregunta, ¿por qué no aplicar los beneficios de una aplicación móvil a esta área, que está limitada a la tecnología al ser útil solo en la versión web? ¿Podría En este proyecto queremos acercar la tecnología de las aplicaciones móviles al área de Vinculación con el Medio, brindando todos los beneficios que otorgan las nuevas tecnologías que tenemos al alcance de nuestra mano.

1.1 Descripción del proyecto

El trabajo de título se realizó en modalidad Proyecto de Software, el cual consiste en el desarrollo de un sistema para solucionar una problemática presente, aportando con avances y nuevas innovaciones.

Empezando por el análisis de la actualidad, se describe la problemática, principales puntos de mejora y aportaciones que se pueda ofrecer a la forma actual en que se realiza seguimiento a las actividades de vinculación con el medio. Junto con esto, se analizaron las opciones y se eligieron las adecuadas

según los requerimientos y necesidades del sistema. El sistema está enfocado a solventar las necesidades de innovar y renovar un sistema que si bien funciona, tiene debilidades, tardaba mucho tiempo en la aprobación de las iniciativas.

Se decidió tener un sistema flexible de consultas a la base de datos (ORM) con el fin de hacerla flexible, además de separar frontend y backend, para que cada uno funcione de forma separada y sea versátil a cambios.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar una APP móvil para android en primera instancia (si bien no se habrá probado para IOS se presume que será funcional) que contenga herramientas para optimizar los procesos de la plataforma SISAV de la UTEM, dando la opción de que los usuarios puedan tener, al alcance de su smartphone, las funciones más importantes del sistema SISAV.

1.2.2 Objetivos específicos

- Analizar la versión actual de SISAV, para observar sus procesos actuales y deficiencias.
- Elaborar un sistema intuitivo, respetando patrones principales de la aplicación actual.
- Desarrollar una interfaz que sea amigable para el usuario del sistema, según los patrones y el perfil de la institución.

- Elaborar un manual técnico para el correcto manejo de futuras actualizaciones y continuación del proyecto para futuros estados.

1.3 Alcances y limitaciones

1.3.1 Alcances

- El sistema contempla las siguientes actividades:
- Se registran los movimientos normales de un administrador y de un usuario normal de la plataforma SISAV, con el fin de hacer seguimiento a las actividades cotidianas y más importantes.
- Se realizará una aplicación móvil para android extendible para IOS, que contenga estas actividades cotidianas, sólo las más importantes, definidas mediante el método anterior.
- El sistema permitirá a los usuarios realizar las tareas cotidianas desde su smartphone.

1.3.2 Limitaciones

- Se acotan los sistemas de seguridad, dando pie a un nuevo trabajo de título, que se dedique plenamente al trabajo sobre la seguridad de la información.
- Se acotan los perfiles de usuarios al avance, se hará un único perfil de usuario capaz de realizar todas las acciones posibles.

1.4 Metodología

El proyecto, al ser realizado en un tiempo definido y no necesitar de un equipo grande de desarrollo, tendrá cabida dentro del margen de la metodología ágil, siendo su proceso de desarrollo guiado por el modelo en espiral.

El desarrollo o modelo en espiral es un enfoque de desarrollo de software que puede ser considerado como una respuesta a los inconvenientes del desarrollo en cascada. El modelo en espiral describe el ciclo de vida de un software por medio de espirales, que se repiten hasta que se puede entregar el producto terminado. El desarrollo en espiral también se conoce como desarrollo o modelo incremental. El producto se trabaja continuamente y las mejoras a menudo tienen lugar en pasos muy pequeños.

Una característica clave del desarrollo en espiral es la minimización de los riesgos en el desarrollo de software, lo que podría resultar en un aumento de los costes totales, más esfuerzo y un lanzamiento retrasado. Estos riesgos son contrarrestados por el enfoque incremental, haciendo primero prototipos, que luego pasan al menos una vez, por las fases de desarrollo de software. El desarrollo en espiral es genérico y puede combinarse con otros métodos de desarrollo clásicos y ágiles, por lo que también se denomina modelo o desarrollo de segundo orden.

Imagen 1. Scrum en pocas palabras

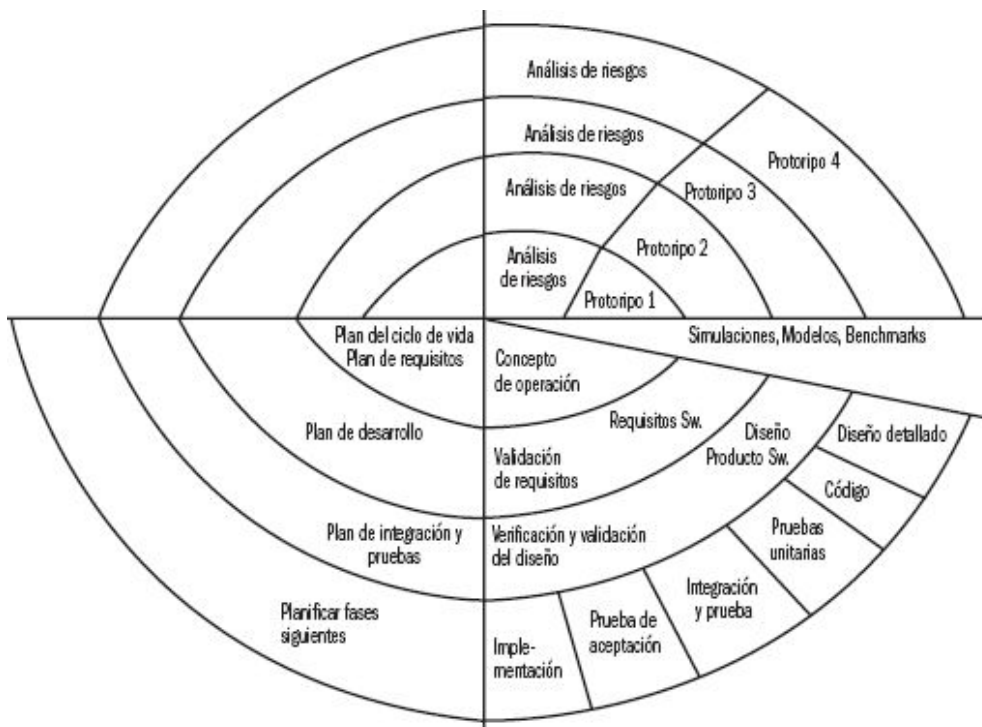


Fuente: <http://jparad.blogspot.com/2012/08/modelo-espiral.html>

1.4.1 Cómo funciona

El modelo de desarrollo en espiral se caracteriza por los siguientes ciclos (también cuadrantes):

Imagen 2. Método de espiral.



Fuente: https://www.goconqr.com/es-ES/p/6786850-Tipos-de-ciclos-de-Vida-mind_maps

- Objetivo y determinación alternativa: Los objetivos se determinan conjuntamente con el cliente. Al mismo tiempo, se discuten posibles alternativas y se especifican las condiciones marco (por ejemplo, sistemas operativos, entornos y lenguajes de programación).
- Análisis y evaluación de riesgos: Se identifican y evalúan los riesgos potenciales. También se evalúan las alternativas existentes. Los riesgos son registrados, evaluados y luego reducidos utilizando prototipos,

simulaciones y softwares de análisis. En este ciclo, existen varios prototipos como plantillas de diseño o componentes funcionales

- Desarrollo y prueba: Los prototipos se amplían y se añaden funcionalidades. El código real es escrito, probado y migrado a un entorno de prueba varias veces hasta que el software pueda ser implementado en un entorno productivo.
- Planificación del siguiente ciclo: El siguiente ciclo se planifica al final de cada etapa. Si se producen errores, se buscan soluciones, y si una alternativa es una mejor solución, se prefiere en el siguiente ciclo.

La fuerza impulsora más importante del desarrollo en espiral es el análisis y la evaluación de riesgos, por ende cualquier riesgo que amenace el proyecto debe ser identificado desde el principio. El progreso del proyecto depende decisivamente de cómo se puedan eliminar los riesgos y es considerado exitoso sólo cuando no hay riesgos.

El objetivo del ciclo es producir un producto en continua mejora, o sea la aplicación se perfecciona constantemente, el modelo en espiral es incremental, pero no necesariamente repetitivo y las repeticiones ocurren sólo cuando los riesgos, errores o conflictos amenazan el proyecto. Si esto pasa, el producto tiene que pasar por un ciclo de nuevo, llamado una iteración o repetición[1].

[1] fuente: https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm

1.4.2 Actividades del proyecto: Scrum

Cada actividad del proyecto será guiada por la metodología Scrum, con el fin de ir validando cada avance con el cliente, con el fin de acomodarnos totalmente a sus necesidades y satisfacer el producto final a cabalidad. El sector del desarrollo de software es el principal representante de este tipo de metodología ágil. Se trata de planificar tus proyectos en pequeños bloques o Sprints, e ir revisando y mejorando el anterior.

Esta metodología es utilizada mayormente para trabajar en grupos de 6 a 8 personas, sin embargo, se plantea según la división de actividades en distintas personas. Entonces, dado el carácter individual del proyecto, se realizará solo por una persona, que tomará las funciones de desarrollador, a la vez de documentador y de comunicador directo con el cliente, ayudando entonces a tener comunicación fluida. En caso de necesitar sugerencias, alternativas, o consejos respecto al desarrollo del proyecto, habrá un profesor guía, que ayudará en las dudas o consultas que se presenten durante la ejecución.

El proceso de desarrollo contempla las siguientes fases que gestionan el proyecto de manera iterativa de acuerdo con la metodología mostrada en la Imagen 1 seguido por la descripción de cada una:

Imagen 3. Scrum en pocas palabras



Fuente: <https://www.diegocalvo.es/metodologia-scrum-metodologia-agil/>.

1.4.2.1 Fase 0- Product Backlog

Se comenzará con la primera reunión, llamada Sprint Planning Meeting, donde se concentra la elaboración de un archivo que recoge el conjunto de tareas o requerimientos y funcionalidades del sistema.

1.4.2.2 Fase 1 - Sprint Backlog

Se definirá un documento con las actividades y tareas que se realizará, generando un cronograma con las horas de trabajo asignado, las reuniones programadas propias de la metodología y las fechas de entregables.

1.4.2.3 Fase 2 - Sprint

Período en el que se realizarán todas las acciones pactadas en el Sprint Backlog y supone entregas parciales para ir testeando el producto final.

Este proceso se repetirá hasta que todos los elementos del Backlog hayan sido entregados.

Dentro de esta fase se encuentra en cada iteración las sub-fases de prueba y entrega, las que serán revisadas hasta que se complete el producto final propuesto.

1.4.2.4 Fase 3 - Producto

En esta última fase se hará entrega del producto terminado, dando paso a la presentación de trabajo de título.

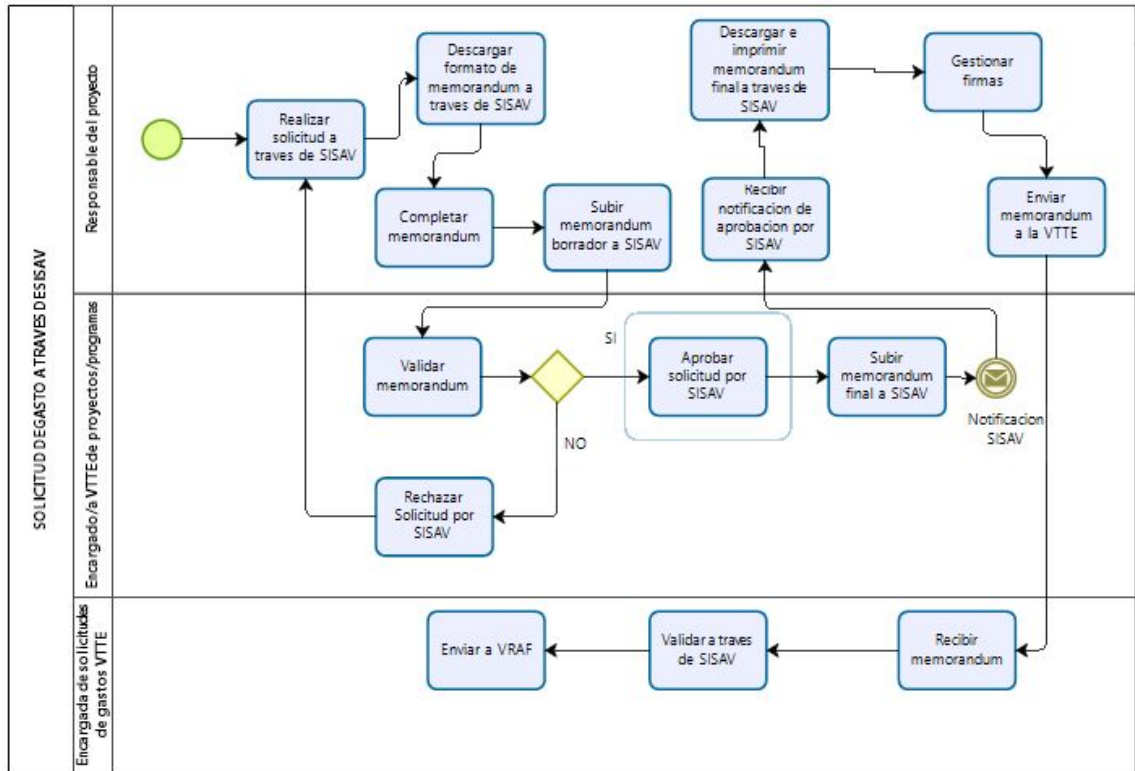
1.5 ANÁLISIS Y RESULTADOS

1.5.1 Diagnóstico

La plataforma SISAV es un espacio solo en versión Web para que académicos puedan presentar proyectos o actividades educativas, con el fin de que sea evaluado por miembros evaluadores, establecidos por cuatro áreas competentes: Evaluador de VTTE (Vicerrectoría De Transferencia Tecnológica y Extensión), evaluador de VRAC (Vicerrectoría Académica), evaluador externo de VEDP (Vinculación con el entorno disciplinar profesional) y evaluador externo de proyectos A+S. Una vez aprobados los proyectos se mantiene un seguimiento completo según las fechas previamente establecidas, hasta el fin del mismo.

A través de este diagrama de flujo, describiremos entonces, el proceso completo la plataforma SISAV:

Diagrama 1: Diagrama BPMN del sistema SISAV.



Fuente: Archivos internos plataforma SISAV, elaborado por desarrollador del área.

1. Se genera una solicitud de actividad a través de la plataforma SISAV. La petición es realizada por el encargado del proyecto.
2. Se descarga el formato de memorándum, a través de la plataforma SISAV. La descarga es realizada por el encargado del proyecto.
3. Se procede al llenado del formato, para luego subir el borrador a la plataforma por parte del encargado del proyecto.
4. Luego, viene la etapa de validación del memorandum por parte de el o la encargada VTTE de proyectos/actividades,, presentándose dos casos:

5. Se acepta la solicitud de parte de SISAV, continuando con la siguiente etapa.
6. Se rechaza la solicitud de parte de SISAV, volviendo al punto 1. Está la posibilidad de volver a llenar el formulario y repetir todos los pasos anteriores.
7. Al ser aprobado el borrador, se sube el memorándum final y se notifica al encargado del proyecto.
8. El encargado del proyecto procede a descargar e imprimir el memorándum final aprobado.
9. Se gestionan las firmas necesarias, por parte del encargado del proyecto.
10. El encargado del proyecto debe enviar el memorándum a la VTTE, con el fin de proceder a la obtención del financiamiento solicitado para la realización de las actividades ya aprobadas.
11. El encargado de las solicitudes de gastos recibe el memorándum, lo valida a través de la plataforma sisav y procede a enviarlo a VRAF.
12. Luego de esto, se hace seguimiento a las actividades según estaban planeadas, con el fin de verificar el estado de cada una.

1.5.2 Sistema SISAV actual

1.5.2.1 Login

Imagen 4. Login SISAV.



Sistema de Seguimiento de Actividades de Vinculación con el Medio

PLATAFORMA DE GESTIÓN DE VINCULACIÓN CON EL MEDIO

Ingresar con Pasaporte.UTEM

ejemplo@utem.cl

Ingrese Contraseña

[Reportar problema](#)

Entrar

Acceso unico a traves de pasaporte.UTEM

Fuente: <https://sisav.utem.cl/>.

1.5.2.2 Acceso principal (Perfil visita)

Imagen 5: Página principal SISAV.



Fuente: <https://sisav.utem.cl/>.

1.5.2.3 Opciones menú general

Aca podemos ver el menú con sus diferentes opciones, cabe mencionar que no todos los usuarios tendrán acceso a estas opciones, sin embargo, el usuario como privilegios máximos tendrá estos accesos, por lo que solo queda filtrar según nivel de usuario.

Imagen 6: Vista menú principal SISAV.



Fuente: <https://sisav.utem.cl/>.

- Iniciativas VcM: Área de propuestas de Vinculación con el Medio.
- Programas: Programas del área de Vinculación con el Medio.
- UET: Proyectos del área de Vinculación con el Medio.
- Extensión: Proyectos culturales.
- POA: Plan Operativo Anual de Vinculación con el Medio.
- Actividades: Actividades en fase de ejecución.

- Financiamiento: Memorándum de Actividades de Vinculación con el Medio.

1.5.3 Oportunidades de Mejora

Para un buen manejo del sistema, en la actualidad se requiere acceder desde la página web en modo responsive, ya que al acceder desde una pantalla de menor tamaño se hace imposible ajustarla, teniendo que navegar de forma manual para un mejor desempeño, imposibilitando el buen uso.

La opción de observar y descargar documentos como el memorándum solo se permite desde la plataforma SISAV, desde un aparato móvil no es posible.

Durante el proceso de análisis de una iniciativa existe una parte especificada en el punto 7 del modelo BPMN en la figura 1 (página 29-30), correspondiente a la recolección de firmas de parte del comité evaluador que resulta un trámite muy engorroso, tomando por lo general alrededor de dos a tres semanas. La recolección de firmas por parte del encargado del proyecto hace referencia a la acción del encargado, de solicitar personalmente la firma de los cuatro evaluadores para así proceder a la solicitud de recursos para solventar la actividad. por ende, depende completamente de la disponibilidad de cada uno de los evaluadores, además de la del encargado del proyecto. Sin la presencia de alguna de las partes se hace imposible la aprobación. por ende podemos observar una gran oportunidad de mejora en este aspecto, ya que de hacer más dinámica esta actividad, el tiempo del proceso total bajaría considerablemente, estipulando entonces plazos pertinentes y más acorde a lo aceptable.

Para mejorar esta situación en primera instancia se propone desarrollar una versión móvil de la aplicación SISAV, con el fin de ayudar al usuario a tener una mejor experiencia de utilidad, dando facilidades de acceso a la plataforma, acomodando la página al tamaño de un aparato móvil estándar, sea tablet, smartphone, etc.

Para mejorar el tiempo de los procesos se propone realizar la firma digital, permitiendo a los funcionarios evaluadores firmar de forma digital la aprobación de las actividades, para que al momento de que el encargado del proyecto imprima el documento, este ya se encuentre firmado, listo para enviar a la próxima fase de facilitación de recursos para el desarrollo de la actividad.

Se podrá subir respaldos a las propuestas utilizando periféricos del teléfono móvil, como la cámara, y cargarlos directamente en la aplicación.

1.5.4 Beneficios cuantitativos y cualitativos al implementar App SISAV

- El principal beneficio al implementar el nuevo sistema es la reducción de tiempo, ya que al ser firma digital bastaría con un solo clic de parte del personal evaluador para aprobar o rechazar una actividad.
- Ahorra el problema de disponibilidad del comité evaluador al revisar los antecedentes y firmar los documentos solicitados para el normal desarrollo de las actividades.
- Aumenta las opciones de trabajo para los usuarios de todos los niveles de la plataforma SISAV, ya que solo se podía operar el sitio web y desde una computadora de escritorio. Con esto se da paso para que en

cualquier lugar y a cualquier hora se puedan aprobar o rechazar los proyectos.

- Facilita la revisión de documentos desde el smartphone, por lo que se hace menos tedioso para los usuarios.
- El proceso actual de subida de respaldos de gastos destinados a la actividad por parte del encargado del proyecto consiste en fotografiar o escanear documentos, enviar las fotos a un ordenador para luego poder subir a la plataforma. Con el nuevo sistema bastará con fotografiar los documentos con la cámara del smartphone y subir directamente a la plataforma a través de la aplicación móvil.
- Se podrá ver las actividades en proceso por parte de los encargados de proyecto y en qué situación se encuentran sus solicitudes desde su smartphone.

Capítulo 2: Formulación y delimitación del problema en estudio

2.1 Descripción de la organización

2.1.1 Historia

La Universidad Tecnológica Metropolitana, más conocida por su acrónimo como UTEM, es una institución de educación superior pública estatal de Chile, creada el 30 de agosto de 1993, a partir del Instituto Profesional de Santiago, heredero del Instituto Politécnico de la Universidad de Chile. Es una de las dieciséis universidades del Consorcio de Universidades del Estado de Chile, y pertenece al Consejo de Rectores de las Universidades Chilenas.

Posee cuatro sedes, todas ellas ubicadas en la Región Metropolitana. Su casa central se encuentra en la comuna de Santiago, siendo parte de las antiguas casonas de la calle Dieciocho de la ciudad, declarada Zona Típica por el Consejo de Monumentos Nacionales del país.

La plataforma SISAV corresponde al sistema de seguimiento de las actividades de vinculación con el medio.

2.1.2 Vinculación con el medio

Vinculación con el medio se define como una acción bidireccional, en que se incorporan las perspectivas de ambos actores (institución y comunidad) en busca de beneficios mutuos entre la academia y su entorno.

2.1.3 Vinculación con el medio: UTEM

Como uno de los propósitos estratégicos de la Universidad Tecnológica Metropolitana, la vinculación con el medio está plasmada en su misión, en la que se propone la “generación, transferencia, aplicación y difusión del conocimiento en las áreas del saber que le son propias, para contribuir al desarrollo sustentable del país y de la sociedad de la que forma parte”.

Asimismo, en el Plan de Desarrollo Estratégico 2011-2015, dos de las cinco definiciones estratégicas corresponden a este ámbito:

- Fortalecer la vinculación con el medio y su retroalimentación en los procesos académicos.
- Fortalecer y construir capacidades para el desarrollo de los postítulos y educación continua.

Tras estas orientaciones, en 2013 se aprueban la política, los procesos y los procedimientos (Resolución N° 02342/2013) de Vinculación con el Medio, los que recogieron la experiencia de la gestión institucional en esta materia hasta entonces, así como las recomendaciones que hiciera la Comisión Nacional de Acreditación en 2011.

En marzo de 2014, la Universidad recibió el acuerdo de la CNA que señalaba la acreditación del área voluntaria de Vinculación con el Medio, lo que significó un respaldo a los avances logrados y al fortalecimiento de este ámbito en la institución.

Así, desde entonces, se han reforzado los instrumentos descentralizados de VcM, como la planificación anual de proyectos desde los Centros de Facultad y los programas de transferencia tecnológica; a la vez que se crean otras herramientas, como la metodología de aprendizaje y servicios que enfatiza la retroalimentación a la docencia; el Consejo Asesor Social Empresarial (CASE), que refuerza el vínculo con y desde el entorno; y un conjunto de iniciativas para el seguimiento de egresados.

2.1.4 Estructura Organizacional

La estructura organizacional es fundamental en todas las empresas, define muchas características de cómo se va a organizar, tiene la función principal de establecer autoridad, jerarquía, cadena de mando, organigramas y departamentalizaciones, entre otras. En este caso, la estructura organizacional está enfocada en quienes participan en la plataforma de seguimiento actualmente, diferenciándose por niveles de permisos que tiene cada usuario:

2.1.5 Permisos según niveles de usuario

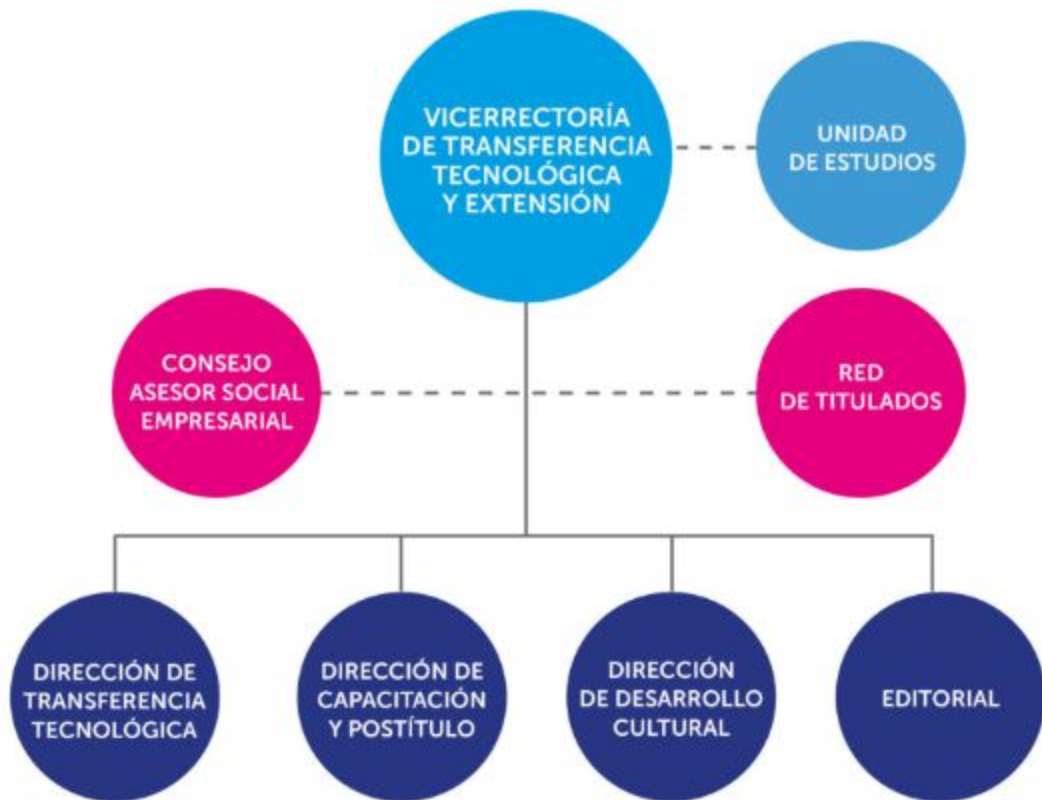
- Nivel de usuario 1 (Directores): Puede ver las actividades en desarrollo, las que están por aprobar, las aprobadas y las rechazadas. También edita todo a nivel facultad.
- Nivel de usuario 2 (Decano): Ve todo, pero no puede editar nada.

- Nivel de usuario 3 (Académicos): Puede realizar solicitudes de actividades, ver estados, subir documentación y solicitar evaluaciones. Apunta a los solicitantes de actividades.
- Nivel de usuario 4 (Evaluadores): Puede ver la documentación adjunta, los presupuestos, los fondos destinados a las actividades junto con actividades en desarrollo, las que están por aprobar, las aprobadas y las rechazadas.
- Nivel de usuario 5 (Programadores): Puede agregar usuarios, editar entradas y administrar bases de datos, nivel de usuario de programador.
- Nivel de usuario 6 (DEMO): Puede ver lo básico, nivel de usuario de visita.

Cada nivel da autorización a cada usuario para poder interactuar con las distintas fases de la plataforma, teniendo cada quien el nivel adecuado según su puesto. Esto cumple un rol fundamental a la hora del orden y de buena gestión del sistema ya que no cualquiera puede ver los proyectos, como también, hay cargos superiores que deben revisar cada movimiento con el fin de que exista transparencia a la hora de financiar actividades y destinar los recursos a actividades que correspondan.

2.1.6 Organigrama de la organización

Figura 1: Organigrama VTTE



Fuente: <https://vtte.utem.cl/vtte/>

2.1.7 Usuarios de la plataforma

Los usuarios que participan en el sistema de seguimiento son los siguientes:

- DIRECTOR DE ESCUELA (Usuario 1)
- DIRECTOR DE DEPARTAMENTO (Usuario 1)
- DIRECTOR DE CENTRO O FACULTAD (Usuario 1)

- DIRECTOR VTTE (Usuario 1)
- USUARIO GLOBAL (Usuario 6)
- ADMINISTRADOR (Usuario 5)
- USUARIO CONSULTOR (Usuario 2)
- ACADÉMICO (Usuario 3)
- DECANO (Usuario 2)
- EVALUADOR UE-VTTE (Usuario 4)
- EVALUADOR VRAC (Usuario 4)
- EVALUADOR EXTERNO A+S (Usuario 4)
- EVALUADOR EXTERNO VEDP (Vinculación con el entorno disciplinar profesional) (Usuario 4)

2.2 Descripción del problema en estudio

El proyecto consiste en el desarrollo e implementación de una aplicación móvil que permita a los usuarios del SISAV tener una mejor experiencia en los procesos actuales accediendo desde sus Smartphones, ahorrando tiempo y dando dinamismo y tecnología, además de sencillez. Junto con esto, se propondrán mejoras a la versión actual de internet.

La principal idea del proyecto es dar a los usuarios más facilidades de acceso a la plataforma, poniéndolo a la altura correspondiente de una universidad tecnológica, aprovechando al máximo las actuales tecnologías que están al alcance de todos.

2.3 Requerimientos

Para realizar este trabajo de título será necesario contar con recursos de hardware, software y humanware, los que se mencionan de manera general a continuación:

2.3.1 Hardware

Será necesario contar con un computador (de escritorio o notebook) con conexión a internet para desarrollar el sistema en un servidor local. Junto con esto, se necesitará un smartphone, que hará las fases de prueba a tiempo real, con el fin de ejecutar e ir probando.

2.3.2 Software

Los principales recursos de software son:

- Sistema operativo Windows 10.
- MySQL Workbench.
- TypeScript.
- Editor de texto Visual Studio Code.
- NodeJS.
- Ionic framework en su versión 4.12.0 o 5, que son compatibles.

2.3.3 Humanware

El único recurso de este tipo para el desarrollo del proyecto es el estudiante de la asignatura de trabajo de titulación.

Capítulo 3. Alternativas de solución

3.1 Descripciones y evaluación de las alternativas

Imagen 7: Esquema descripción y evaluación de alternativas.



Fuente: <https://ingenioempresa.com/analisis-de-alternativas/>

3.1.1 Descripción

La toma de decisiones es el proceso mediante el cual la persona debe escoger entre dos o más alternativas. En general, en sistemas informáticos se enfoca en decidir si se creara un sistema nuevo o se utilizara uno existente, pero en este caso ya que se propuso ampliar las actuales herramientas de un sistema ya desarrollado enfocaremos este ítem en la elección de herramientas de trabajo. Algunos beneficios de analisis de tomas de decisiones son los siguientes:

- Efectos futuros: Tiene que ver con la medida en que los compromisos relacionados con la decisión afectarán al futuro del sistema. Una decisión que tiene una influencia a largo plazo, puede ser considerada una decisión de alto nivel, mientras que una decisión con efectos a corto plazo puede ser tomada a un nivel muy inferior. En este caso, la estructura solicitada por el cliente tiene que responder a un sistema separado e independiente en su estructura, o sea, debe estar separado el FRONT del BACKEND, y de la base de datos. Por ende la principal decisión fue usar lenguajes separados independientes y almacenados en distintos lugares con el fin de que sean altamente versátiles.
- Reversibilidad: Se obtiene la capacidad de regresar atrás si la decisión no fue la adecuada, conocido también como derecho a retracto. También da la oportunidad al cliente para realizar cambios existiendo tiempo de reacción.

3.1.2 Ponderación peso según alternativas

- **ORM: Usar o no usar**

Object-Relational mapping, o lo que es lo mismo, mapeo de objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

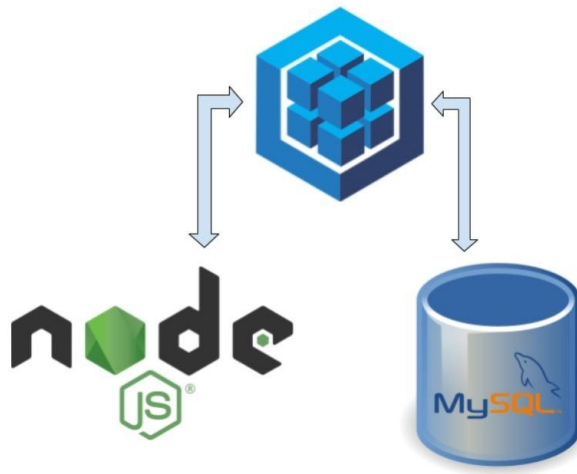
Para este proyecto existe la opción de utilizar un ORM para las consultas a la base de datos, o por el contrario usar lenguaje SQL nativo, con las consultas de forma tradicional. Para poder ver la mejor opción debemos poner puntos de comparación entre ambas, ubicando las ventajas y desventajas de cada una.

Puntos de comparación:

- Facilidad de uso y velocidad: Debemos escoger entonces la opción más fácil para el programador y más ágil con el fin de tener una rápida respuesta y que sea de agrado.
- Abstracción de la base de datos utilizada: Dentro de los requerimientos se encuentra la fácil adaptabilidad a cambios en el tipo de base de datos y nombres. Debe ser versátil y resistente a cambios.
- Seguridad de la capa de acceso contra ataques: Debe ser una herramienta segura.
- Aprendizaje: Debe ser un lenguaje conocido o estar preparado para utilizarlo de manera rápida. Ayuda también en este punto la documentación.

- Rendimiento: Debe estar preparado para el intercambio de datos y modificaciones múltiples.

Imagen 8: Nodejs, sequelize, Mysql.



FUente: <https://medium.com/@ahmedcharef/sequelize-orm-with-node-js-koa-mysql-foreign-keys-a3097ab91a8>

- Unificación del proyecto: Separar o no separar FRONT - BACKEND

La separación de FRONT y Backend es bastante común para proyectos grandes por la fácil delegación de actividades y encargados, sin embargo también se utiliza para proyectos que varían a través del tiempo y tienen muchas actualizaciones o cambios de tecnología, ya que al tenerlos separados resulta más fácil el cambio de plataformas.

Los puntos de comparación en este caso son pocos, ya que no existen ventajas o desventajas en desempeño, en utilidad, en carga de datos ni ninguna característica de ese tipo, si no que apunta a la administración, edición y manipulación del código de mejor forma

3.1.3 Ponderaciones de todas las alternativas según criterios de evaluación

- Tabla comparativa con ORM vs sin ORM:

Tabla 1: Comparación con ORM vs sin ORM

Punto de comparación	Con ORM	Sin ORM (SQL puro)
Manejo del lenguaje, dificultad a la hora de programar	Complejo si no se domina el lenguaje.	Sencillo, lenguaje SQL.
Velocidad	Rápido con intercambio de datos moderado.	Rápido incluso con grandes intercambios.
Abstracción de la base de datos utilizada	No depende del tipo de base de datos, compatible con MySQL, Sqlite, Postgresql, entre otras.	Depende del tipo de base de datos, no es flexible. No resiste cambios de base de datos.
Seguridad	Segura. Se ocupa automáticamente de higienizar los datos que llegan, evitando posibles ataques de inyección SQL y similares.	Segura, sin embargo es vulnerable ante algunos tipos de ataques.
Reutilización del código	Permite la fácil reutilización del código ya que se desliga de un formato definido.	No ayuda con la reutilización.
Duplicidad	Se elimina la duplicidad de código.	Alta probabilidad de gastar recursos llamando dos o más veces al mismo dato.

Fuente: Realización propia.

Se concluye entonces que la mejor opción es utilizar un ORM, y es el punto de la abstracción el clave, ya que dentro de los requerimientos existe la solicitud de un sistema flexible ante cambios de base de datos, y adaptable.

- Ventajas y desventajas de utilizar FRONTEND - BACKEND separados

Ventajas:

- Se tiene un orden correcto de desarrollo, ya que se permite realizar primero el Backend con todas las funciones posibles, para luego dar paso al Frontend, que con los datos ya proporcionados puede consumir y dar libre uso.
- Permite que tanto el frontend como el backend funcionen independientes del otro, es decir, al ser IONIC un framework libre puede consumir datos de cualquier Backend que cumpla los formatos, independiente del lenguaje que se utilice en este, y por otro lado, el Backend puede proporcionar datos a cualquier frontend compatible con su estructura, independiente del lenguaje.
- Se pueden editar a la par, y trabajar de forma paralela, por ende es más manejable.
- Pueden encargarse equipos por cada una de las partes, dando pie a la rápida mejora y solución de errores.

Desventajas:

- Resulta difícil la conexión entre ambas, por lo que se debe tener buena comunicación durante el proyecto y las ediciones que pueda tener.
- Dividir las características en dos partes entre dos personas aumentará la carga de trabajo general.
- Como desventaja general hace que existan expertos en cada una de estas disciplinas, por lo que al editarlas se necesita alguien con dominio FULL STACK o dos personas que se puedan complementar.

Como conclusión, al ser solo un programador que maneja todo el proyecto, se decide separar FRONTEND y BACKEND, asumiendo que las principales desventajas apuntan al trabajo en equipo, que no es el caso.

3.2 Beneficios de la solución

- Aumenta la visibilidad y el tráfico online

Hoy en día es muy importante mantenerse actualizados y observar las oportunidades de mejora que ofrecen las nuevas tecnologías, junto con esto son muchísimos los usuarios que están atentos a las nuevas tendencias en aplicaciones y buscan las últimas novedades que están a su alcance para facilitarles las cosas. Por ende, ofrecer nuevas formas de ingresar a la plataforma genera un mejor desplazamiento y más opciones de accesibilidad.

Mejora la imagen y seriedad del área, apuntando a la importancia de la vinculación con el medio para la acreditación de la universidad.

- Ayuda a la agilización de los procesos

Las firmas electrónicas generan un ahorro de tiempo significativo según el proceso tradicional actual. Además, si una aplicación es ingeniosa y resolutive, puede ser muy satisfactoria para los usuarios.

- Vía de comunicación directa

Tener una app desarrollada permite informar de estados de peticiones, estadísticas y novedades en todo momento desde la app móvil.

- Una solución sin límites temporales

Una empresa puede cerrar sus oficinas a las 7 de la tarde, pero no por eso deja de “atender” a sus clientes. Mediante una aplicación, los usuarios pueden adquirir las firmas y aprobaciones o rechazos en cualquier momento del día, en todos los días de la semana.

- Adaptación al entorno móvil

Al no existir una web responsive, su uso se limita solo a ingresar desde un ordenador, por lo que aplicarlo permite abrir nuevas áreas de trabajo.

Capítulo 4. Requerimientos del sistema

4.1 Requerimientos de Información

Para su correcto funcionamiento como principal requerimiento se presenta una Base de datos actualizada con los elementos básicos para inicio de sesión y antecedentes de la plataforma. Para un correcto funcionamiento de la aplicación y actualización de datos en tiempo real se solicita subir la aplicación a un servidor, en el que ya funciona la plataforma SISAV.

4.2 Requerimientos Técnicos, de Control y Funcionales

Para el buen funcionamiento de la aplicación se necesitan ciertos requerimientos a nivel de plataforma, los cuales serán divididos en técnicos, de control y funcionales.

- Requerimientos Técnicos:
 - Se necesita un servidor en el que esté la base de datos actualizada a tiempo real, o sea, la misma desde donde se consumen los datos de parte de la página web actual.

- Servidor con windows 7, 8 o 10, con 1gb de ram y espacio de 15 gb de memoria en el disco duro.
- Requerimientos de control:
 - Se necesita un desarrollador encargado del buen funcionamiento, corrección de errores y encargado de las actualizaciones, encargado además de la documentación de la aplicación. Se debe también dedicar a la capacitación del personal, con el fin de utilizar de buena forma la aplicación.
- Requerimientos funcionales:
 - Versión 11.6 node, importante mantener la versión.
 - cordova CLI.
 - npm install (para generar la carpeta node_modules y las instalaciones de NPM necesarias).
 - instalación IONIC versión 5.

4.3 ESTUDIO DE FACTIBILIDAD

4.3.1 Factibilidad Técnica

Consideraciones de orden tecnológico que deba realizar la organización. Se enfoca en obtener un entendimiento de los recursos tecnológicos disponibles actualmente y su aplicabilidad a las necesidades que se espera tenga el

proyecto. En el caso de tecnología informática esto implica una evaluación del hardware y software.

4.3.2 Económica

El propósito del estudio de viabilidad económica, es determinar los beneficios económicos del proyecto o sistema propuesto para la organización, en contraposición con los costos. Generalmente incluye un análisis costo beneficio, el cual se prepara como parte del caso de negocio de un proyecto.

4.3.3 Operativa

Se enfoca en el grado con el cual este encaja con el entorno de negocios existentes y objetivos estratégicos, respecto al cronograma, fecha de entrega, cultura organizacional y procesos de negocio.

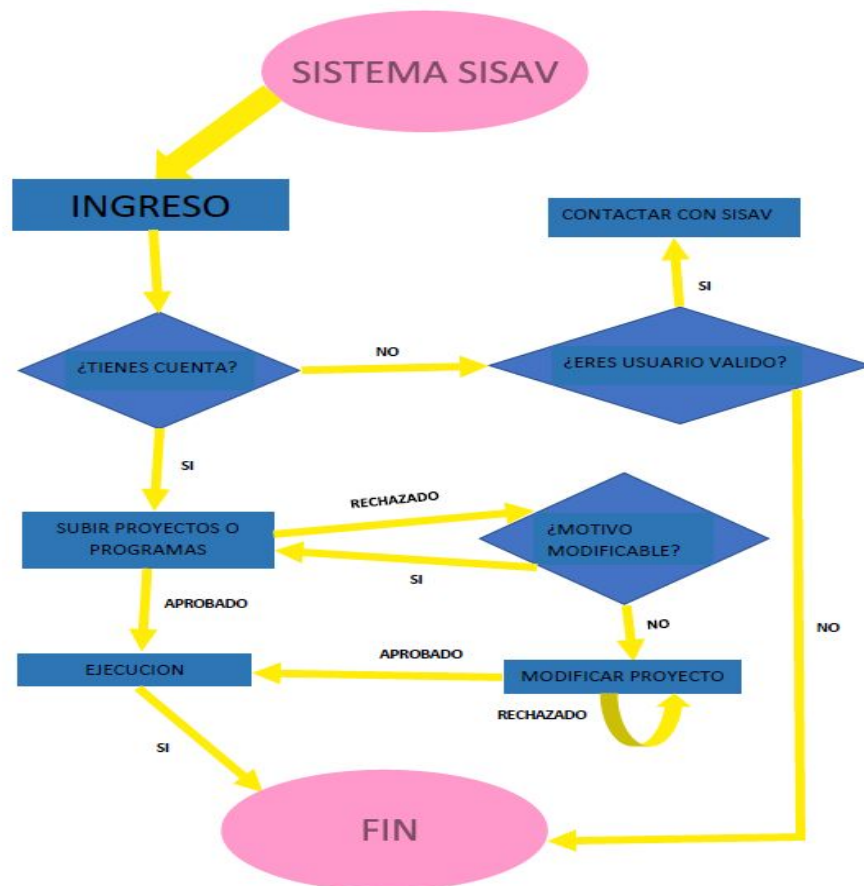
Capítulo 5. Diseño

5.1 DISEÑO LÓGICO

5.1.1 Diagramas

5.1.1.1 Diagrama de flujo de datos

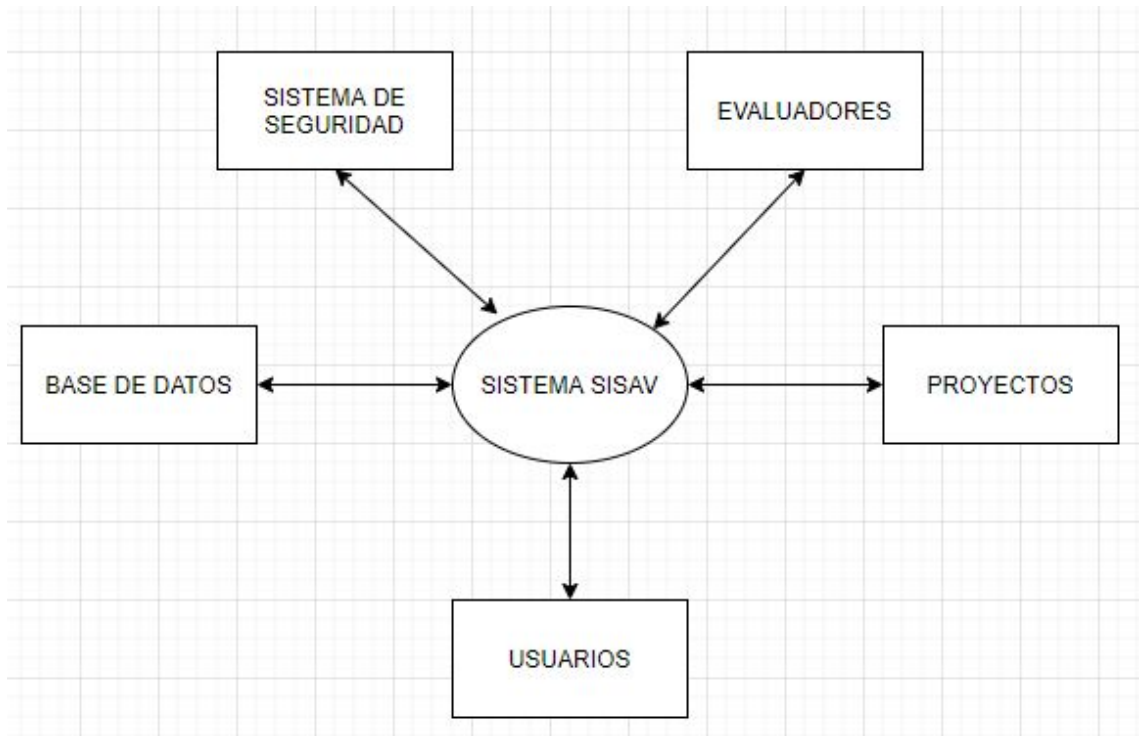
Diagrama 2: Diagrama flujo de datos.



Fuente: Elaboración propia.

5.1.1.2 Diagrama de contexto

Diagrama 3: Diagrama de contexto



Fuente: Elaboración propia.

5.1.2 Diseño de interfaces y descripción

Fase de Análisis:

- Reuniones con responsables para establecer una visión clara del sistema a diseñar: Fue de suma importancia entablar reuniones con el solicitante con el fin de establecer los requisitos que debe tener el nuevo sistema a aplicar. En este caso se estableció que la aplicación debe contar con los

colores institucionales adecuados (azul oscuro y blanco) y con las pestañas correspondientes a la actual versión de escritorio del sistema.

- Inclusión de tareas relativas a usabilidad en el plan del proyecto: Se establece también que debe ser un sistema de fácil uso, con herramientas simples que logre el objetivo principal.

Búsqueda de productos competitivos:

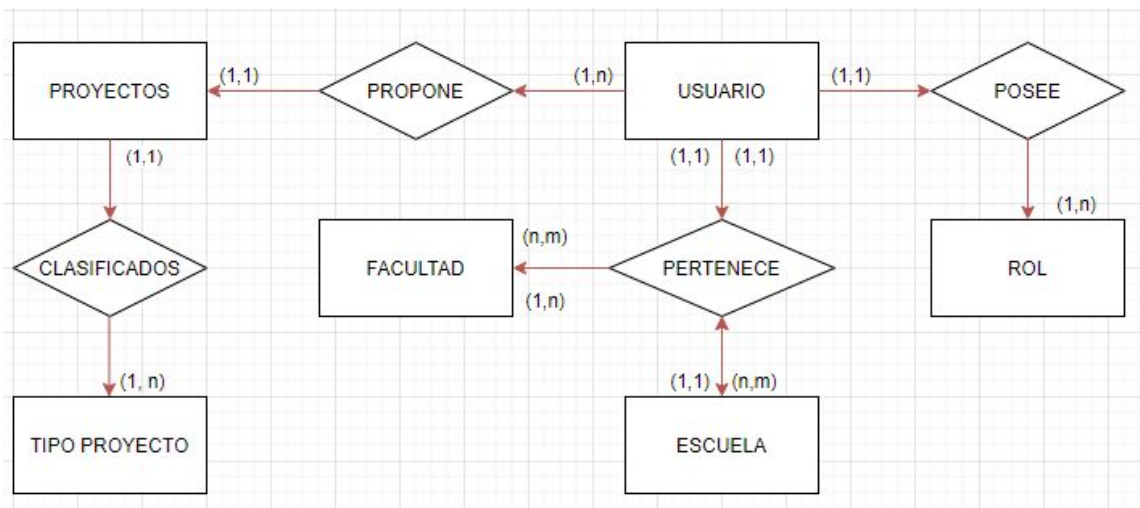
- Crear perfiles de usuario: el desarrollo de los perfiles de usuario es de gran importancia en el proyecto, ya que se debe verificar muy bien quienes tendrán acceso a las herramientas de la aplicación móvil.

5.2 Diseño físico

5.2.1 Modelamiento

5.2.1.1 Modelo Entidad Relación

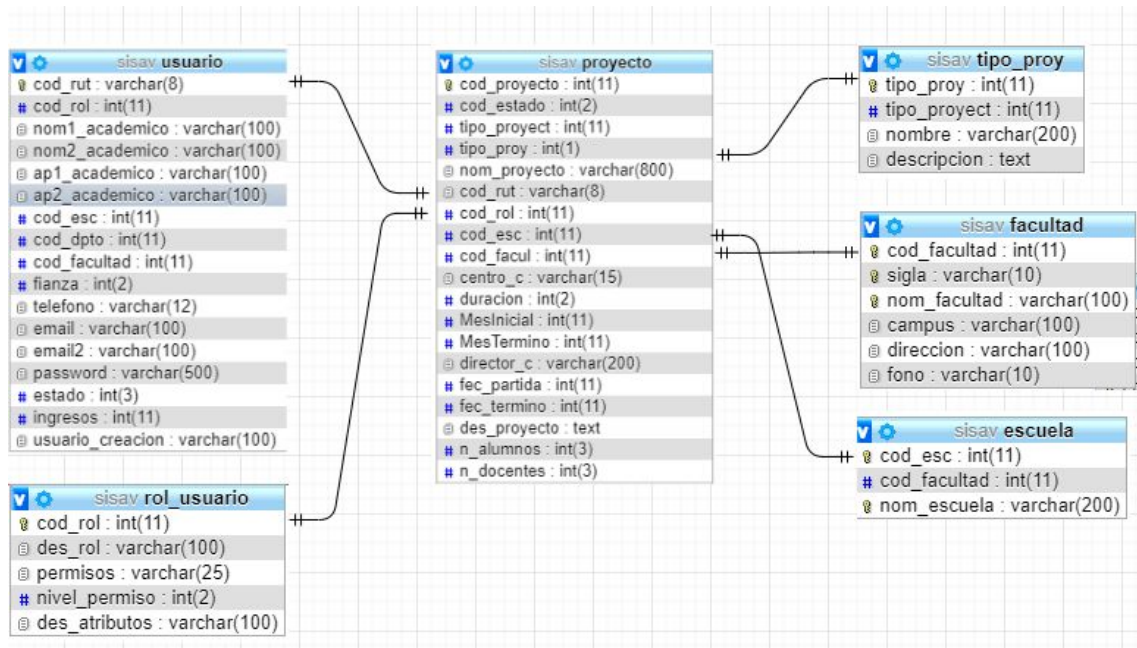
Diagrama 4: Modelo entidad relación



Fuente: Elaboración propia.

5.2.1.2 Modelo Relacional

Diagrama 5: Modelo relacional.



Fuente: Elaboración propia.

5.2.2 Análisis y Evaluación de Riesgos

En el momento de implementar un sistema los riesgos pueden ser muchos, desde un mal desarrollo y cobertura de los requerimientos hasta un mal manejo de la instalación en sí del sistema o la capacitación del personal. En esta parte mencionaremos los riesgos a los que está expuesto este proyecto y cuales son los planes de mitigación de los mismos.

El proceso de análisis de riesgo genera habitualmente un documento al cual se le conoce como matriz de riesgo. En este documento se muestran los elementos identificados, la manera en que se relacionan y los cálculos

realizados. Este análisis de riesgo es indispensable para lograr una correcta administración del riesgo. La administración del riesgo hace referencia a la gestión de los recursos de la organización. Existen diferentes tipos de riesgos como el riesgo residual y riesgo total así como también el tratamiento del riesgo, evaluación del riesgo y gestión del riesgo entre otras.

Como se describe en el BS ISO/IEC 27001:2005, la evaluación del riesgo incluye las siguientes actividades y acciones:

- Identificación de los activos: Activos se refiere al recurso de valor empleado en una empresa. En este caso se refiere principalmente a la información y datos de los usuarios y de los programas.
- Identificación de los requisitos legales y de negocio: Para el correcto funcionamiento se deben cumplir los estándares básicos de seguridad de la información, los cuales están estipulados en la organización.
- Valoración de los activos identificados: En este caso se refiere principalmente a la importancia de la protección de los datos tanto de la universidad como de los usuarios de esta.
- Identificación de las amenazas y vulnerabilidades importantes: Verificar principalmente el cumplimiento de los parámetros de seguridad mínimos, tales como uso de token y encriptación de contraseñas. Nuestra principal amenaza es el ataque externo con el fin de obtener datos estadísticos de los programas y proyectos realizados en la utem, como la información de los usuarios.

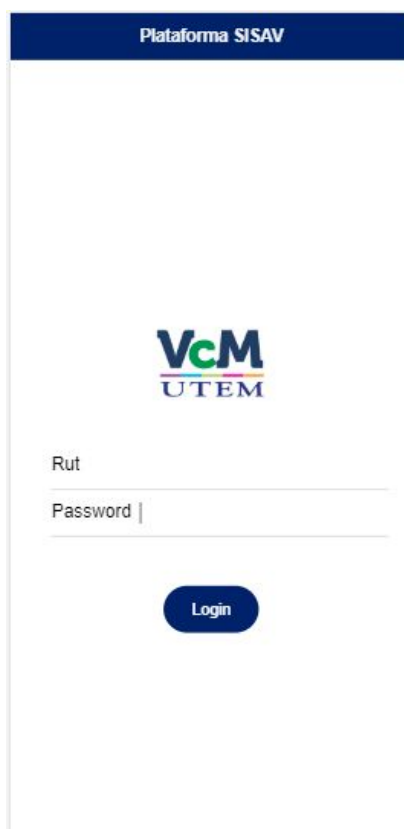
Después de efectuar el análisis debemos determinar las acciones a tomar respecto a los riesgos residuales que se identificaron. Las acciones a realizar en el caso de SISAV son:

- Controlar el riesgo: Detectar el lugar vulnerable de la plataforma y analizar los datos que estén en riesgo.
- Eliminar el riesgo: Generar una solución ante el problema anterior, con el fin de evitar que suceda una fuga de información.
- Compartir el riesgo: Se deja nota del plan de contingencia efectuado como prevención, con el fin de que no vuelva a ocurrir.
- Aceptar el riesgo: Cuando está controlado el problema y solucionadas las vulnerabilidades se procede a dar cuenta del riesgo que se acaba de evitar.

5.3 Vistas

5.3.1 Login

Imagen 9: Login

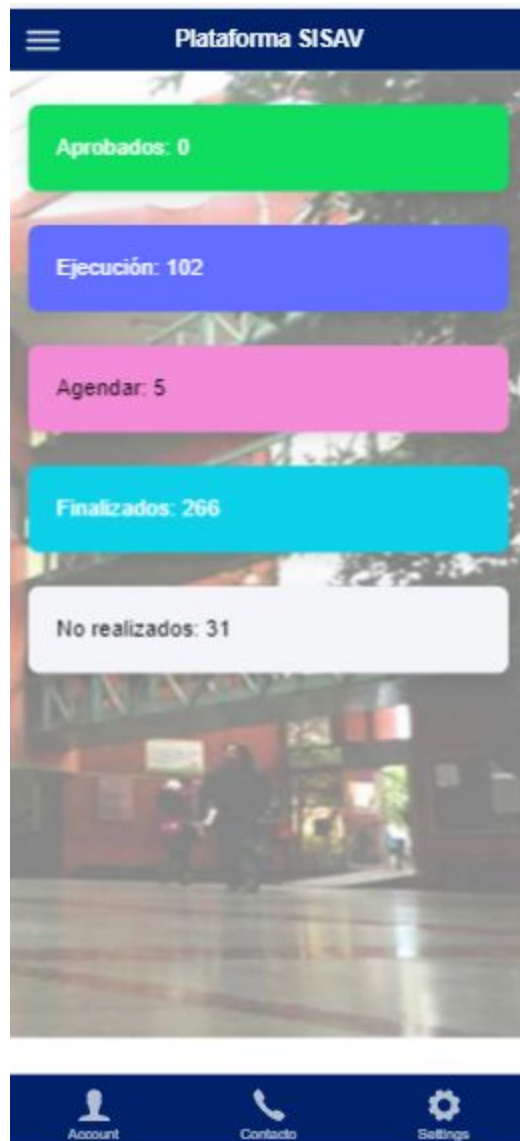


The image shows a login interface for the 'Plataforma SISAV'. At the top, there is a dark blue header bar with the text 'Plataforma SISAV' in white. Below the header, the page is white. In the center, there is a logo for 'VcM UTEM'. The logo consists of the letters 'VcM' in a stylized font, with 'V' in blue, 'c' in green, and 'M' in blue. Below 'VcM' is the word 'UTEM' in a smaller, blue, sans-serif font. Under the logo, there are two input fields. The first is labeled 'Rut' and the second is labeled 'Password |'. Both fields are empty and have a light gray border. Below the input fields, there is a dark blue button with the word 'Login' in white text.

Fuente: Elaboración propia.

5.3.2 Acceso principal

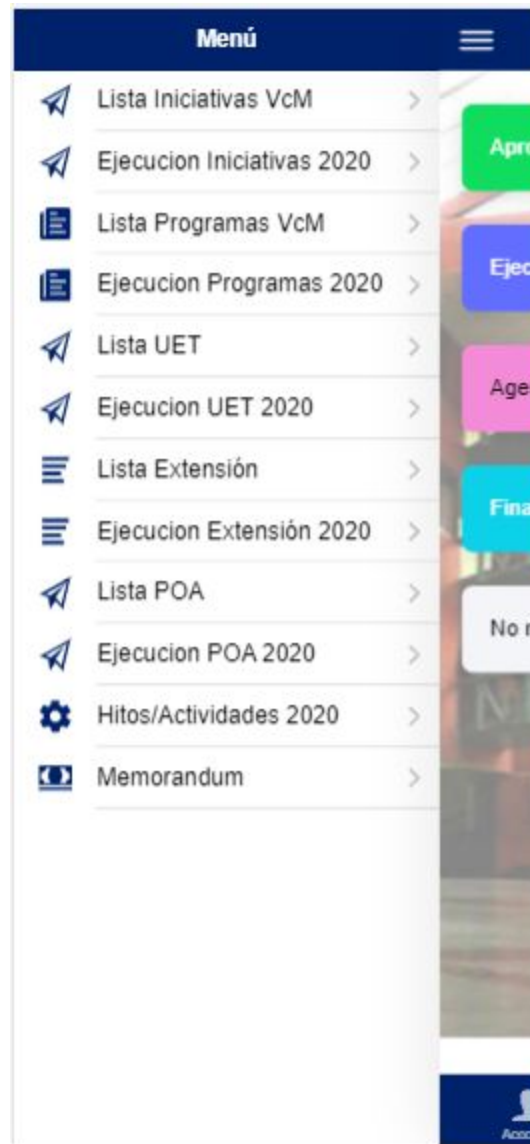
Imagen 10: Acceso principal.



Fuente: Elaboración propia.

5.3.3 Menú de opciones (IPHONE)

Imagen 12: Menú de opciones 1.

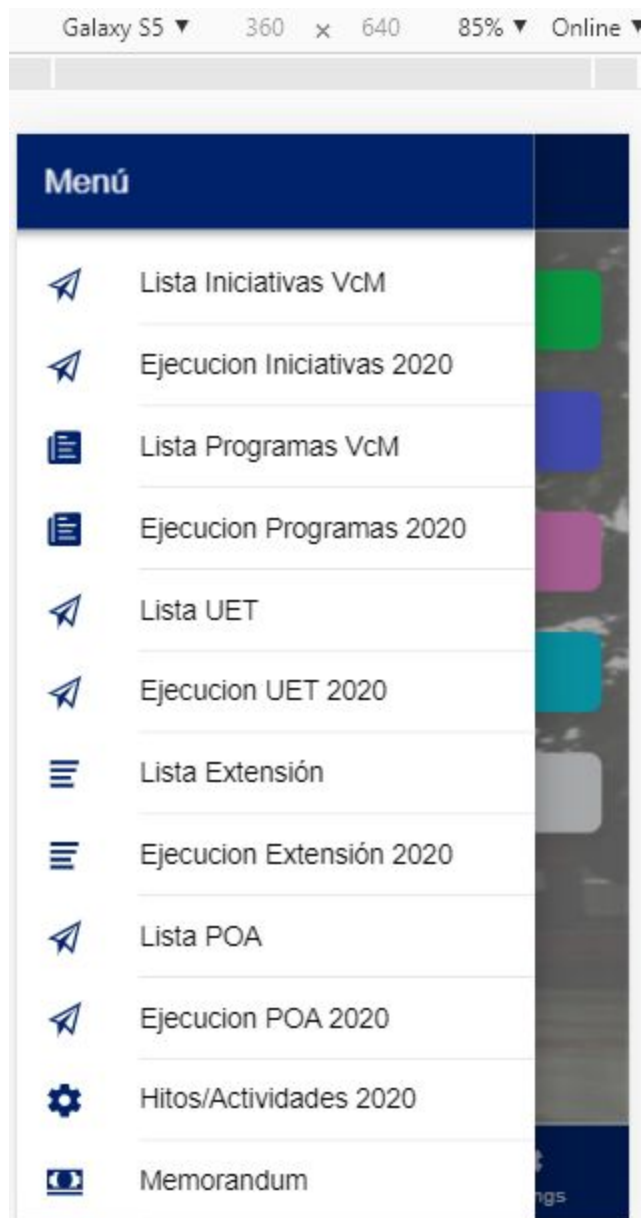


Fuente: Elaboración propia

Podemos ver que el menú hace que la pantalla principal se deslice junto con el menú de opciones, función que viene adquirida con el sistema operativo IOS.

5.3.4 Menú de opciones (tipo galaxy S5)

Imagen 13: Menú de opciones 2.

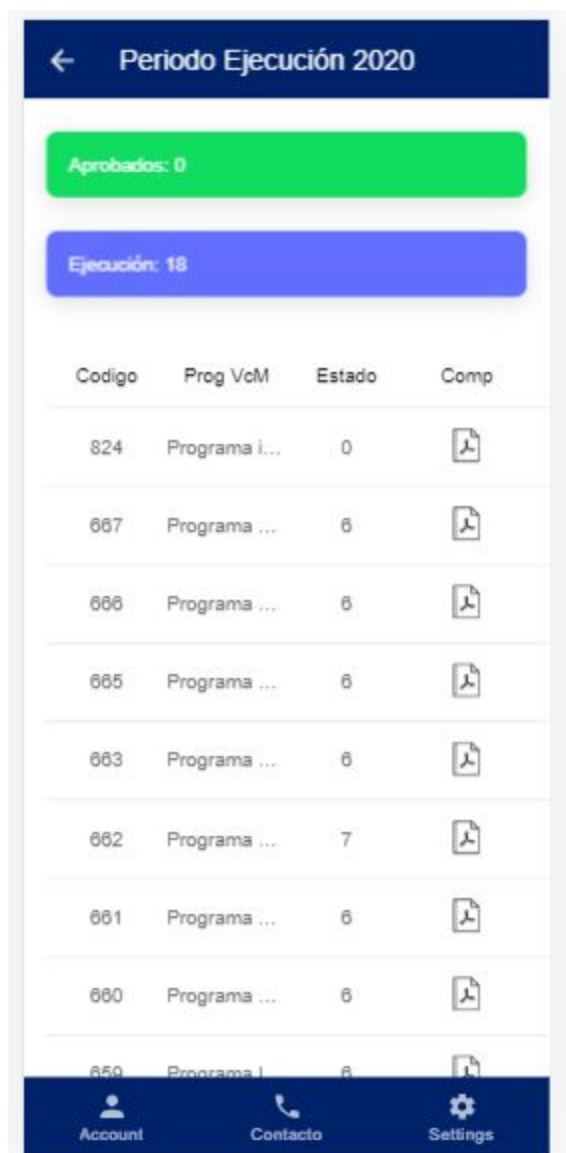











Fuente: Elaboración propia.

En este caso, se observa como el menú de opciones se sobrepone a la pantalla principal, dando el efecto característico del sistema operativo Android.

5.3.5 Vista lista programas

Imagen 14: Vista de programas.



Código	Prog VcM	Estado	Comp
824	Programa i...	0	
667	Programa ...	6	
666	Programa ...	6	
665	Programa ...	6	
663	Programa ...	6	
662	Programa ...	7	
661	Programa ...	6	
660	Programa ...	6	
659	Programa I...	6	

Fuente: Elaboración propia.

5.3.6 Vista lista proyectos

Imagen 15: Vista lista de proyectos.

← Periodo Ejecución 2020

Aprobados: 0

Ejecución: 84

Codigo	Prog VoM	Estado	Comp
822	Ciclo de C...	4	
821	VINCULAC...	6	
820	Publicación...	6	
819	OBSERVA...	6	
818	Workshop ...	6	
817	Evaluación...	6	
816	Actividad d...	6	
814	Aprendizaj...	6	
813	Muestra de	6	

Account Contacto Settings

Fuente: Elaboración propia.

5.4 Resultados obtenidos

- Como principal resultado se obtiene una aplicación móvil, capaz de mostrar los datos estadísticos actuales en cuanto a los proyectos y programas que se desarrollan en la actualidad, junto con ver los estados de estos, sean propuestas, estén aceptadas en ejecución o finalizadas. Lo importante de este resultado final es que el proyecto queda realizado en su base y abierto a recibir futuras actualizaciones que complementan constantemente las necesidades de SISAV.
- En esta aplicación se puede acceder a través de un login funcional, pero que no cuenta con la encriptación implementada por dos motivos:
 - No se puede acceder al tipo de encriptación utilizada en la base de datos actual, ya que es confidencial y si se aplica debe ser con supervisión del respectivo personal competente. Sin embargo la estructura para su aplicación queda instaurada, siendo sencillo de aplicar en una actualización futura.
 - No refleja el objetivo del trabajo de título. Desde el comienzo se plantea que la seguridad de la información no abarca este campo, sin embargo quedan todas las terminales preparadas para la aplicación de la seguridad correspondiente, tomando fácilmente parte de una actualización sencilla.
- El login cuenta con token, lo que permite limitar la inactividad y cantidad de veces que se puede abrir a la vez una cuenta. Además, evita la multiplicidad de datos y peticiones instantáneas.

- Da la opción de mostrar un PDF resumen de todos los documentos asociados a un proyecto o programa con el fin de ver y validar su realización.
- Queda aplicado solo el nivel total de permisos, quién puede hacer todo y deshace todo, pensando en que en una futura actualización se puedan fijar los niveles de usuario correspondientes.
- Presenta la opción de cambiar de estado una propuesta de proyecto, es decir, se puede aprobar o rechazar un proyecto. Sin embargo cabe mencionar que por ahora la opción de generar el PDF validado queda inconclusa, pero con la opción de implementación fácil, al estar habilitadas las opciones dando pie a la externalización de la función, con el fin de no recargar la aplicación.

Capítulo 6. Conceptos básicos

6.1 App móvil

Una app móvil es una aplicación de software pensada para dispositivos móviles y tabletas. Aunque hay quienes lo remontan a los primeros móviles de los años 90, lo cierto es que el término app empezó a usarse en el año 2008, con el lanzamiento, en primer lugar, de App Store de la empresa Apple y, en segundo lugar, de Android Market, conocido hoy por Google Play. También

cabe destacar, en los orígenes de las apps móviles, el primer SDK para desarrolladores de Android.

Para usar una aplicación móvil se necesita un smartphone o algún otro aparato móvil con acceso a internet, teniendo en cuenta que no todas las aplicaciones funcionan en todos los aparatos móviles. Dependen entonces del sistema operativo y el tipo de aplicaciones que corresponde a ese aparato. Los sistemas operativos móviles Android, Apple, Microsoft y BlackBerry tienen tiendas de aplicaciones que operan en línea en las cuales se puede buscar, descargar e instalar las aplicaciones. Algunos comerciantes minoristas también operan tiendas de aplicaciones en internet. Se debe usar una tienda que le ofrezca las aplicaciones que funcionen con el sistema operativo de su aparato. Para establecer una cuenta, es posible que tenga que suministrar el número de una tarjeta de crédito, especialmente si va a descargar una aplicación que no es gratis.

6.2 Bases de datos

Se entiende por “Base de Datos” al conjunto de datos estructurado y almacenado de forma sistemática con objeto de facilitar su posterior utilización. Una base de datos puede constituirse con cualquier tipo de datos, incluyendo los de tipo puramente espacial tales como los que se utilizan en un SIG, así como, por supuesto, datos numéricos y alfanuméricos como los que constituyen la componente temática de la información geoespacial

Los elementos clave de la base de datos son esa estructuración y sistematicidad, pues ambas son las responsables de las características que hacen de la base de datos un enfoque superior a la hora de gestionar datos.

Las bases de datos en la actualidad nos permiten hacer consultas, listados, depuraciones, manejo de claves, niveles de usuarios, etc. de manera digital, lo que hace que todo se almacene en reducidos espacios físicos, y muchas veces, se alojen en la nube, reduciendo entonces a 0 el espacio. Antiguamente, se utilizaba documentación física como respaldo, ya sean documentos impresos de tablas, archivos a los que se podía acceder de forma manual, tardando mucho en encontrar los datos solicitados, perdiendo fácilmente información, no encontrándola, o por ejemplo, muy fácil de perder toda la información en accidentes tales como incendios, terremotos, errores humanos como el derramamiento de café, entre otros.

Las bases de datos en la actualidad, son de forma digital y nos ofrecen múltiples beneficios. Algunos son:

- Mayor independencia. Los datos son independientes de las aplicaciones que los usan, así como de los usuarios.
- Mayor disponibilidad. Se facilita el acceso a los datos desde contextos, aplicaciones y medios distintos, haciéndolos útiles para un mayor número de usuarios.
- Mayor seguridad (protección de los datos). Por ejemplo, resulta más fácil replicar una base de datos para mantener una copia de seguridad que hacerlo con un conjunto de ficheros almacenados de forma no estructurada. Además, al estar centralizado el acceso a los datos, existe una verdadera sincronización de todo el trabajo que se haya podido

hacer sobre estos (modificaciones), con lo que esa copia de seguridad servirá a todos los usuarios.

- Menor redundancia. Un mismo dato no se encuentra almacenado en múltiples ficheros o con múltiples esquemas distintos, sino en una única instancia en la base de datos. Esto redundará en menor volumen de datos y mayor rapidez de acceso.
- Mayor eficiencia en la captura, codificación y entrada de datos.

Esto tiene una consecuencia directa sobre los resultados que se obtienen de la explotación de la base de datos, presentándose al respecto ventajas como, por ejemplo:

- Mayor coherencia. La mayor calidad de los datos que se deriva de su mejor gestión deriva en mayor calidad de los resultados.
- Mayor eficiencia. Facilitando el acceso a los datos y haciendo más sencilla su explotación, la obtención de resultados es más eficiente.
- Mayor valor informativo. Resulta más sencillo extraer la información que los datos contienen, ya que uno de los cometidos de la base de datos es aumentar el valor de estos como fuente de información.

6.3 Frontend

Frontend es la parte de un sitio web que interactúa con los usuarios, es decir, la parte visual de un sistema. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios, por consiguiente, la principal misión de un Desarrollador Front-End es

conseguir una experiencia de usuario atractiva y dinámica, intuitiva y amena, sencilla y con colores adecuados según lo que se necesita mostrar, a través del desarrollo de los diferentes componentes externos del sitio o aplicación web. Su objetivo es desarrollar todos los elementos visuales de un sitio web, con los que un usuario interactúa.

6.3.1 Desarrollo frontend: Frameworks y Lenguajes

HTML + CSS + JavaScript como conjunto son los lenguajes principales del Frontend, de los que se desprenden una cantidad de frameworks y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios. React, Redux, Angular, Bootstrap, Foundation, LESS, Sass, Stylus y PostCSS son algunos de ellos.

6.3.1.1 HTML + CSS + JavaScript

Como lenguajes y dialectos más populares aparece este conjunto. que son utilizados en diferentes plataformas de desarrollo. Es por esto que solo entraremos a definir solo estas tres:

- HTML: HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto

- CSS: Es el estilo que le aplicamos a cada uno de los elementos creados previamente en el HTML. Por ejemplo, le añadimos a los párrafos un color al texto, una tipografía, un tamaño.

En CSS se usan los “selectores”, que seleccionan (valga la redundancia) etiquetas de HTML, para añadir una serie de propiedades y aplicarle el estilo.

- Javascript: Generalmente (y de forma muy básica) se conoce como la animación que suele aparecer en una web. Por ejemplo: hago click en un párrafo y el párrafo desaparece o cambiar de color. Aunque Javascript realmente es un lenguaje de programación y un “dialecto” de ECMAScript, se usa para diferentes tecnologías como rutinas de navegador (Node JS), frameworks(Angular, React, Vue, Ionic) y animaciones en páginas web con frameworks(Tween Max, jQuery, Mootools).

En empresas del tipo agencia de publicidad técnicamente se pide más el uso de jQuery. Mientras que en startups, y bancos se comienzan a pedir más conocimiento de frameworks.

6.4 Backend

El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no se ven, como, por ejemplo, la comunicación con el servidor. Se utiliza para la comunicación entre el frontend y la base de datos,

prácticamente es todo el funcionamiento, las consultas, las solicitudes, modificaciones, es el encargado de hacer todos los movimientos.

Algunas de las funciones que se gestionan en la parte del back-end son:

- El desarrollo de funciones que simplifiquen el proceso de desarrollo.
- Acciones de lógica.
- Conexión con bases de datos.
- Uso de librerías del servidor web (por ejemplo para implementar temas de caché o para comprimir las imágenes de la web).
- Seguridad de los sitios web que gestiona y optimizar al máximo los recursos para que las páginas sean ligeras.

CAPÍTULO 7: MANUAL DEL PROGRAMADOR

7.1 Introducción

El manual del programador, también conocido como Manual Técnico juega un papel importante dentro del sistema debido a que luego de instalar el sistema y ponerlo en producción, se tiene la tarea de darle mantenimiento para que el sistema continúe siendo operacional. En este caso el manual del

programador se llevará a cabo para las futuras actualizaciones llevadas a cabo por los próximos estudiantes en tesis, o los miembros que sean dispuestos por las autoridades competentes. La idea de esto es explicar paso a paso los elementos del código del proyecto, para ayudar a esta labor.

En este caso, como las personas que aplicaran las actualizaciones o nuevas funciones al sistema son nuevos programadores que no han tenido contacto antes con el proyecto se es necesario contar con una herramienta de manual técnico que permita aprender fácilmente cómo está integrado el sistema desde el punto de vista técnico, presentando claramente cada uno de los procesos del sistemas y su interrelación para formar el sistema completo. Además de indicar cada uno de los datos o información que se almacena en la base de datos del sistema, sus relaciones y las transformaciones que sufren los datos para convertirse en información.

7.2 Sistema de bases de datos

En este proyecto se utilizó una base de datos ya creada y utilizada por “SISAV”. Base de datos creada en “mysql”. Por motivos de seguridad se omiten los nombres de estas tablas, sin embargo solo se accedera a 4 tablas que nombraremos con alias. Una tabla dedicada a las actividades o proyectos del área, con sus características, otra tabla para los académicos con el fin de entrar al sistema y obtener los permisos según nivel de usuario, otra tabla dedicada a el estado de las actividades, sean en ejecución, realizadas, no realizadas, finalizadas, etc. y la última para conocer los roles de usuario para el login. Para poder administrar y consumir datos de esta base de datos se utiliza el motor de bases de datos “XAMPP”, quien nos proporciona un servidor local que permite acceso a la base de datos.

7.3 Back-end (API REST)

El sistema backend aplicado en este proyecto fue realizado con NODEJS, y las consultas a la base de datos fueron gestionadas por el modelo de programación “SEQUELIZE”. Sequelize es un ORM para Nodejs que nos permite manipular varias bases de datos SQL de una manera bastante sencilla, entre estas bases de datos podemos encontrar: mysql, sqlite, postgres, mssql.

7.3.1 Controladores (Controllers)

En este espacio van ubicadas las funciones que nos retornan los resultados esperados. Técnicamente lo que hacen es hacer las consultas a las tablas de bases de datos y guardar los resultados para su posterior utilización. Dentro de la carpeta “Controllers” se encuentra el archivo “index.js”, encargado de exportar los resultados obtenidos divididos por tipo de tabla base de datos, entonces, cada vez que se añada un nuevo controlador, se debe exportar en el archivo “index.js”.

7.3.2 Modelos (Models)

En esta carpeta se dibujan las estructuras de las tablas de bases de datos, con los nombres de cada campo (debe llevar el mismo nombre que en la base de datos) y su tipo, correspondiente al dato ingresado. Ejemplo:

Nombre: DataTypes.STRING

En este caso se hace referencia al elemento “Nombre” de una tabla de base de datos, que es de tipo “STRING”.

Así entonces se dibuja la estructura completa de la tabla de base de datos, permitiendo al controlador obtener una respuesta completa a las funciones (peticiones) realizadas. Se debe hacer mención especial al elemento “index.js” de esta carpeta.

El “index.js” funciona como fuente de comunicación entre sequelize y la base de datos, y es donde se conecta por ende nuestro backend con la base de datos. De este archivo no se debe cambiar nada, ya que se genera automáticamente al iniciar el proyecto siendo una de las ventajas de utilizar “SEQUELIZE” para el desarrollo.

Este archivo “index.js” también nos ayuda con la estructuración de las características de la base de datos. Esto se realiza mediante un archivo llamado “config.json” ubicado en la carpeta llamada “CONFIG”.

7.3.3 Config (config.json)

En este archivo se encuentran los parámetros para la conexión a la base de datos. Este archivo nos proporciona la facilidad de cambiar los parámetros o tipo de base de datos (mysql, postgres, etc.) solo cambiando el nombre respectivo, ya que “Sequelize” realiza el trabajo de transformar el resto de parámetros para su libre utilización. Ejemplo:

```
"development": {  
  "username": "nombre_de_usuario",  
  "password": "contraseña_ingreso_a_BDD",  
  "database": "Nombre_de_la_BDD",
```

```
"host": "127.0.0.1",  
"dialect": "mysql",  
"operatorsAliases": false  
},
```

Como podemos ver, cambiando el dialecto se pueden utilizar bases de datos distintas, tales como “Postgres, sqlite, mssql”. También, cambiar el nombre de la base de datos para consumir desde otro sector, o cambiar el host, que en este caso es el Localhost.

Como principal beneficio se tiene que estos cambios solo se realizan en esta parte y se aplican a todo el proyecto. Este archivo “config.json” se genera solo, al iniciar el proyecto.

7.3.4 Rutas (Routes)

Las rutas se encargan de dar un link propio a cada una de las funciones realizadas en los controladores. Es decir, cada vez que se solicite la realización de una función se debe llamar al link correspondiente definido en la ruta. Así, se exportan fácilmente los resultados a cualquier lugar del proyecto donde se soliciten. Se pueden poner todas las rutas en un solo archivo, sin embargo se recomienda crear un archivo por cada modelo que tengamos.

7.3.5 App (app.js)

Archivo encargado de generar la pantalla principal de respuesta. El backend al momento de ejecutar, llama a este archivo. que contiene las rutas a exportar y los métodos de solicitudes que se utilizan, tales como “GET, PUT, DELETE,

POST”, etc. Junto con otros métodos que por defecto el sistema utiliza para su correcta ejecución.

7.3.6 www.js

Archivo ubicado en la carpeta “BIN” encargado de crear y conectar al servidor, en este caso localhost. Se seleccionan parámetros por defecto al generar el proyecto.

7.4 FRONTEND

Para la realización del frontend de nuestro proyecto se utiliza IONIC, quien además de proporcionarnos vistas responsivas nos da la posibilidad de ejecutar a través de aplicación móvil. Tiene la ventaja de ser independiente al backend, ya que solo se conecta a una API-REST, sin importar de qué tipo sea. Si bien a IONIC no le importa el tipo de API-REST a la que se conecta es importante aclarar que siempre es necesario tener una API-REST para el funcionamiento, es decir, no se puede conectar directamente IONIC con una base de datos cualquiera.

7.4.1 Imágenes (Carpeta imágenes)

Contiene las imágenes que se utilizarán en todo el proyecto, tanto iconos, como imágenes estadísticas e imágenes de fondo.

7.4.2 SRC

Contiene todo el proyecto. La sigla “src” significa “Origen” y se genera automáticamente al iniciar el proyecto. Es importante que se mantenga este nombre, puesto que al ejecutar llama a esta carpeta.

7.4.3 Theme

Ya dentro de la carpeta “src” se encuentra la carpeta “theme”. Esta carpeta contiene el archivo “variables.scss”, importante para el diseño de las vistas, ya que es acá donde se declaran los colores, sus códigos respectivos y su respectivo nombre para poder utilizarse en las vistas, específicamente en los archivos en HTML.

7.4.4 Environments

Carpeta generada automáticamente, para ayudar con la conexión a la API que se consumirá. Específicamente ayuda a escribir la contraseña que muchas veces contienen las API's, por motivos de seguridad.

7.4.5 Assets

Se utiliza esta carpeta para declarar iconos que podamos utilizar durante el proyecto, pero principalmente se utilizó para declarar el menú desplegable de la izquierda. El archivo “menu.json” contiene la estructura del menú, cuantos espacios tiene, los nombres de cada ítem, y el icono que va delante del nombre. Si se quiere añadir o quitar algún dato del menú, se debe realizar acá.

7.4.6 App

Si bien todos los archivos son importantes, este lo es un poco más, puesto que acá se ubican cada una de las vistas y contenido visual del proyecto.

7.4.7 Components

Ya dentro de la carpeta “app” se encuentra la carpeta components, que contiene los elementos que se utilizan en más de una vista. El fin principal de tener esta carpeta es ahorrar líneas de código y declarar sólo una vez estos elementos y no en todas las vistas.

Esta carpeta contiene 3 subcarpetas:

- Header: Es la barra superior que se observa en todas las vistas, una barra estática que sólo cambia el nombre, por lo que se declara como una función utilizable en todas las vistas.
- Menú: Es el menú desplegable de la izquierda. Siempre será igual. La estructura estaba definida, acá se acomoda en código HTML.
- Piepag: Pie de página.

7.4.8 Interfaces

Contiene el archivo “interfaces.ts”. La respuesta de la api que se consume desde cada una de nuestras vistas tiene una estructura, el rol de esta parte del código es acomodar los datos que se envían de la api para que sean legibles por nuestras vistas.

7.4.9 Services

Contiene el archivo “data.service.ts”. Es acá donde se llaman los links que se generan en la API que estamos consumiendo. La respuesta que se genera recibe la estructura o tipo correspondiente según estén declaradas en el archivo “interfaces.ts”.

7.4.10 Pages

Contiene todas las vistas que se necesitan según los requerimientos de la aplicación.

CAPÍTULO 8: Conclusión

Si bien la plataforma existente para la solicitud de actividades y proyectos referente a la vinculación con el medio de la utem era bastante completa, se considera que acercarla a la tecnología móvil es de vital importancia para facilitar los procesos de la misma.

Junto con esto, luego del análisis exhaustivo que se realizó al proceso de negocio se confirma la necesidad de apurar procesos, y se observan claros puntos de mejora que se trabajaron en la aplicación de SISAV.

Se confirma además que la aplicación móvil ayudará a la carga de datos y respaldos de las actividades, ya que se podrá acceder a los periféricos (cámara) de los aparatos móviles para capturar imágenes.

El uso de las nuevas tecnologías de desarrollo de software es de vital importancia para mantener vigente las herramientas de uso cotidiano en las organizaciones, es por eso que involucrar IONIC y NODEJS se justifica, ayudando al fácil acople y acercando a las principales ventajas emergentes del mercado.

En lo personal, resultó gratificante primeramente trabajar en mejoras para la UTEM como institución, aportando al crecimiento y ayuda al proceso de negocio del área. También, conocer nuevas tecnologías de desarrollo significó un gran desafío como estudiante, confirmando la importancia de ampliar el dominio de estas ya que al existir alta competencia en el mercado actual se hace necesaria la flexibilidad personal, que claramente se vio beneficiada en este proyecto.

Trabajar con un cliente real en este proyecto permitió conocer la exigencia del mercado actual, aprender a tomar requerimientos de mejor manera y ayudó a la comunicación, entendiendo al cliente directo, sus necesidades y aplicándolas de correcta forma en la aplicación final, un punto muy importante a la hora de pensar en el mercado laboral actual, ya que se sale con poco rodaje y se cometen muchos errores por poca experiencia.

CAPÍTULO 9: Anexos

9.1 Sistema informático

Un sistema informático es una técnica que permite el almacenamiento y el proceso de información, para lo cual se vale de un grupo de elementos que se relacionan entre sí, asociados al software y hardware. El usuario, quien requiere la información utiliza estas dos fuentes, y quien es también el que en definitiva tiene el control total de lo que sucede en el sistema.

Un sistema informático está compuesto por:

- Componente físico

Básicamente se trata del hardware del sistema informático. Es decir las computadoras, sus componentes internos como memorias, CPU y demás, los periféricos de entrada y salida como módems, impresoras, monitores, y todo aquel dispositivo que se conecte a este hardware. Los componentes lógicos son los que proporcionan la capacidad y la potencia de proceso para que el sistema informático funcione.

- Componente lógico

Este componente se refiere al software del sistema informático, conformado en primera instancia por el firmware, el sistema operativo y el sistema de gestión de datos propiamente dicho. Además se debe contar como parte del software la documentación del mismo y los datos que procesa y gestiona. El software es el encargado de almacenar, procesar y distribuir los datos que se ingresan al mismo.

- Componente humano

También llamado “Humanware”, este componente está conformado por los usuarios, es decir quienes utilizan los dos anteriores componentes. En este sentido, también deben considerarse como “Humanware” a todos aquellos que han participado en el desarrollo del mismo, es decir ingenieros, programadores y analistas de sistemas. El componente humano de un sistema informático es sumamente importante, ya que además de operar dicho sistema, también son los encargados del soporte y mantenimiento técnico.

9.1.1 Clasificación de sistemas informáticos

Los sistemas informáticos en la actualidad se clasifican en un total de 6 tipos, variando de acuerdo al ámbito en el cual se implementan. Alguno de estos son:

9.1.1.1 Sistemas de apoyo a la toma de decisiones

Sistema diseñado con el propósito de ser usado por la gerencia para ayudar en el proceso de tomar una decisión para resolver problemas y con ello poder diagramar las directrices para seleccionar la mejor opción o predecir los futuros escenarios para afrontar nuevos desafíos.

9.1.1.2 Sistema de control de procesos de negocio

Son aquellos sistemas encargados de monitorizar, controlar y gestionar cualquier proceso de industrialización. Se utilizan sensores electrónicos conectados a computadoras para poder hacer un monitoreo directo del proceso que la maquinaria está realizando, con el objetivo de controlar que el mismo se lleve a cabo con total eficacia.

9.1.1.3 Sistemas de colaboración empresarial

Son el tipo de sistema informático más utilizado por empresas alrededor del mundo, ya que le permiten a las compañías a gestionar la gran cantidad de información que circula dentro de la misma.

La particularidad del ERP es que es un sistema informático que no es de uso específico de un nivel puntual de una compañía, ya que pueden brindar servicios a un abanico importante de usuarios en muchas áreas de la empresa.

9.1.1.4 Sistemas de Información Ejecutiva

Sistema informático capaz de proporcionar acceso inmediato a toda la información crítica que produce la empresa, siempre en un formato que pueda ser fácilmente visualizado y comprendido en una simple mirada. Han sido desarrollados con el objetivo de generar todo tipo de datos e información, la cual se caracteriza por ser lo suficientemente compacta, presenta toda la operación de la empresa.

9.1.1.5 Sistemas de procesamiento de transacciones

Se implementan para gestionar todo lo relacionado con el nivel operacional de la organización. Almacena y procesa todas las transacciones que se realizan en la jornada, y son necesarias para el normal funcionamiento de cualquier empresa.

9.1.1.6 Sistemas de Información de Gestión

Sistemas informáticos capaces de recopilar y gestionar datos de múltiples niveles con el propósito de tener un panorama más claro al momento de tomar decisiones empresariales. Alta capacidad para generar reportes para la gestión

operativa y el control total de todas las actividades de proceso de transacciones.

9.2 Frameworks de Javascript

Hay cientos de frameworks de JavaScript, y constantemente van apareciendo nuevos, por lo que se mencionan solo los de interés en el proyecto:

- Angular: Separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones. Aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Entre otras ventajas es modular y escalable adaptándose a las necesidades del proyecto y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es Typescript, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo por ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad.

Ayuda entonces a las futuras actualizaciones de SISAV y a la paralelización.

Por su programación reactiva, la vista se actualiza automáticamente tras realizar los cambios. Además Angular dispone de asistente por línea de comandos para poder crear proyectos base y también se integra bien con herramientas de testing y con Ionic, lo que facilita la creación de web-responsive, es decir, adaptadas a móviles.

- IONIC: SDK (Software Development Kit) de código abierto para el desarrollo de aplicaciones móviles híbridas. La versión original está construida sobre AngularJS y Apache Cordova. Las versiones más recientes, conocidas como Ionic 5 o simplemente “Ionic”, están basadas en AngularJS. Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como CSS, HTML5 y Sass (Syntactically awesome style sheets).

Ionic facilita la creación de aplicaciones móviles con tecnologías web para los programadores web. La plantilla de Ionic permite a los desarrolladores crear aplicaciones móviles en diferentes plataformas que pueden instalarse en teléfonos con Android e iOS.

¿Por qué usar Ionic?

Ionic facilita la creación de aplicaciones híbridas utilizando HTML5 debido a su Framework de código abierto. El uso de Angularjs lo hace perfecto para el desarrollo de aplicaciones interactivas y tiene una gran

gama de herramientas y servicios que hacen que la ejecución de Ionic sea sencilla.

Ionic viene con CLI (Command-Line Interface), que ayuda a los desarrolladores móviles a construir y probar aplicaciones iónicas en cualquier plataforma. Permite a los usuarios desplazarse por miles de listas sin que su rendimiento se vea afectado. Los usuarios pueden crear sus propias aplicaciones, personalizarlas para Android, iOS y desplegarlas.

Ventajas de IONIC

Las principales ventajas que ofrece son:

- Desde una única fuente podremos llegar a las plataformas que soporta este Framework (Android e iOS).
- El desarrollo principal se realiza en HTML junto con CSS y JS, lenguajes muy extendidos y muy utilizados, por lo que facilitará el desarrollo de proyectos de la forma más efectiva aún cuando la plantilla de desarrolladores contenga nuevas incorporaciones.
- Para el caso de aplicaciones híbridas, se obtiene con un único proceso de desarrollo e implementación, una aplicación para Android, iOS y Web.
- Ionic se centra en construir para los estándares web modernos y para dispositivos móviles modernos. Para Android, Ionic es compatible con Android 4.1 y versiones superiores. Para iOS, Ionic admite iOS 7 en adelante. Ionic 2 es compatible con la

plataforma universal de Windows para crear aplicaciones de Windows 10. Ionic Framework, impulsado por AngularJS, es compatible con las aplicaciones BlackBerry 10.

- El uso de Ionic permite crear, construir, y compilar aplicaciones en cualquier plataforma con un solo comando. Por eso se considera un potente CLI.
- El Ionic Creator, herramienta de ionic permite crear Interfaces mediante plantillas, contiene muchas actualizaciones y soporte, por lo que resulta fácil y rápido crear vistas.

Desventajas de IONIC

Las principales desventajas que contiene:

- El rendimiento puede ser ligeramente menor que en aplicaciones desarrolladas de forma nativa, no debería ser un problema al menos que el proyecto sea para la creación de juegos con detallados gráficos u otras aplicaciones que hagan uso de grandes cantidades de recursos, pero no es el caso de SISAV.
- Es una herramienta “joven” y puede ser difícil encontrar módulos compartidos por los usuarios.
- El navegador no siempre da la información correcta sobre como se mostrará en el teléfono y puede haber fallos en las pruebas, debido a que el navegador solo mantiene las características más comunes de los teléfonos.
- Puede ser difícil de integrar con diferentes funcionalidades nativas.

- Las aplicaciones híbridas son más lentas que las aplicaciones nativas.

9.3 Lenguajes BACKEND

Un lenguaje de programación es un conjunto de reglas que le dicen a un sistema informático qué hacer y cómo hacerlo, dando las instrucciones a la computadora para realizar una tarea en particular. Un lenguaje de programación consiste en una serie de pasos bien definidos que la computadora debe seguir estrictamente para producir el resultado deseado.

- Lenguajes de scripting

Diseñados para integrarse y comunicarse con otros lenguajes de programación. Los ejemplos de lenguajes de script utilizados comúnmente incluyen JavaScript, VBScript, PHP entre otros. Se utilizan principalmente en combinación con otros lenguajes, ya sean lenguajes de programación o de marcado. Por ejemplo, PHP, que es un lenguaje de scripting, se utiliza principalmente en combinación con HTML. Todos los lenguajes de scripting son lenguajes de programación, pero no todos los lenguajes de programación son lenguajes de scripting.

Una de las diferencias entre los lenguajes de script y los lenguajes de programación es en términos de compilación. Si bien es una necesidad para compilar una programación, los lenguajes de scripting se interpretan sin compilarse. Es importante tener en cuenta que los lenguajes de script se interpretan directamente desde el código fuente.

JavaScript (Node Js)

Creado por los desarrolladores originales de JavaScript. Lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comandos como Python pueden crear.

Tanto JavaScript como Node.js se ejecutan en el motor de tiempo de ejecución JavaScript V8. Este motor transforma el código JavaScript en un código de máquina más rápido. El código de máquina es un código de nivel más bajo que la computadora puede ejecutar sin necesidad de interpretarlo primero, ignorando la compilación y por lo tanto aumentando su velocidad.

Utilidades de Node.js

Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente. Puede referirse a cualquier operación, desde leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.

La idea principal es usar el modelo de entrada y salida sin bloqueo y controlado por eventos para seguir siendo liviano y eficiente frente a las aplicaciones en tiempo real de uso de datos que se ejecutan en los dispositivos. Node apunta a las aplicaciones de red rápidas, por lo que en aplicaciones de más peso pierde sus principales ventajas.

Funcionamiento de Node.js

En comparación con las técnicas tradicionales de servicio web donde cada conexión genera un nuevo subproceso, ocupando la RAM del sistema y regularmente maximizando la cantidad de RAM disponible, Node.js opera en un solo subproceso, utilizando el modelo entrada y salida sin bloqueo de la salida, lo que le permite soportar miles de conexiones al mismo tiempo mantenidas en el bucle de eventos.

Cuando hay una nueva solicitud se genera un tipo de evento. El servidor empieza a procesarlo y, cuando hay una operación de bloqueo de entrada y salida, no espera hasta que se complete y en su lugar crea una función de devolución de llamada. El servidor comienza en el acto a procesar otro evento y cuando finaliza la operación de entrada y salida, continuará trabajando en la solicitud ejecutando la devolución de llamada tan pronto como tenga tiempo, por lo tanto, el servidor nunca necesita crear más subprocesos o cambiar entre subprocesos, lo que significa que tiene muy poca sobrecarga.

Ventajas de Node.js

- Javascript está incorporado en la plataforma Node.js, haciéndolo fácil de aprender y manejable por programadores de Java.
- Node.js se desarrolla en un entorno de tiempo de ejecución de fuente libre que ayuda en el almacenamiento de creación de proyectos únicos.
- El modelo de entrada y salida impulsado por eventos ayuda en el manejo simultáneo de peticiones.
- El administrador y el usuario incorporan estrategias de codificación similares que desembocan en la creación de abundantes aplicaciones de internet de gran competencia.

- Al ser un modelo de programación de un solo subproceso, ofrece abundantes características y opciones.
- Node.js es la plataforma de software más utilizada en este momento estando por encima en entornos de ejecución y lenguajes de programación como PHP y C presentando un tiempo de ejecución menor.

CAPÍTULO 10: Bibliografía

Michael Abernethy, IBM. (Junio de 2011). ¿Qué es nodejs?

Fuente:<https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html>

Campusmvp. (Marzo de 2017). Ventajas de usar Angular.

Fuente:<https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx>

Desarrolloweb. (Marzo de 2017). IONIC CLI.

Fuente:<https://desarrolloweb.com/articulos/ionic-cli.html>

Campusmvp. (Marzo de 2017). Ventajas de usar Angular.

Fuente:<https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx>

nextech. (Septiembre de 2016). BPMN, usos y guías.

Fuente:<https://nextech.pe/que-es-bpmn-y-para-que-sirve/>

Oracle (página principal). Detalles, datos y significados respecto a bases de datos.

Fuente:<https://www.oracle.com/cl/database/what-is-database.html>

Udemy. Aprendiendo a programar en IONIC, bases de NODEJS.

Fuente: www.udemy.com

SISAV, sistema de seguimiento de actividades de vinculación con el medio.

Fuente: www.sisav.utem.cl