

Trabajo Receptacional



TÍTULO: SIRA ORIENTADA A SERVICIOS.

Estudiante: Miguel Ángel Landa López
Generación: MRySI 2019-2021

Director de Tesis: Dr. Juan Manuel Gutiérrez Méndez | 22 de Enero de 2022
Co-directora: Mtra. Idalí Nieto Jiménez

Dedicatoria

Este trabajo esta dedicado a:

A mis padres Miguel Ángel e Isabel, por el apoyo incondicional, su esfuerzo realizado, y la paciencia que me han brindado durante toda mi vida para que siga desarrollando mi carrera profesional y hoy pueda cumplir un sueño más.

A Mariana, quien es mi compañera de vida y siempre ha estado ahí para apoyarme y motivarme incondicionalmente; y a mi futuro bebé, quien ha sido la última luz de esperanza y amor que me ha motivado para concluir este trabajo.

A toda mi familia quienes hacen de mí una mejor persona.

Agradecimientos.

Al Laboratorio Nacional de Informática Avanzada A. C., por permitirme el desarrollo del presente trabajo y facilitarme las herramientas necesarias.

A mis directores de tesis Dr. Juan Manuel Gutiérrez Méndez y Mtra. Idalí Nieto Jiménez, por la idea del proyecto, el tiempo dedicado, y el apoyo mostrado en todo momento para resolver mis dudas, inquietudes, y proporcionarme toda la información necesaria que me facilitó enormemente el desarrollo del presente trabajo.

A la Dra. Cora Beatriz Excelente-Toledo y al Dr. Eduardo López Domínguez, por sus valiosos consejos y opiniones durante los seminarios de la experiencia de “Trabajo Terminal”, los cuales me permitieron encausar de una mejor manera el desarrollo del mismo.

Al Dr. Saúl Domínguez Isidro, por su tiempo dedicado y cuyas valiosas observaciones y puntos de vista agregaron un mayor valor y profesionalismo al trabajo realizado.

Y por supuesto a innumerables autores, maestros, programadores, compañeros y excompañeros de trabajo, quienes, a través de su contenido, de sus desarrollos, sus enseñanzas, y, sobre todo las experiencias de trabajo y charlas compartidas en este apasionante mundo del desarrollo del software, han hecho de mí un mejor profesionista.

ÍNDICE.

1. INTRODUCCIÓN	1
1.1 PLANTEAMIENTO E IDENTIFICACIÓN DEL PROBLEMA	2
1.2 OBJETIVO GENERAL Y ESPECÍFICOS	3
1.2.1 <i>Objetivo General</i>	3
1.2.2 <i>Objetivos Específicos</i>	3
1.3 JUSTIFICACIÓN.....	3
1.4 ALCANCES Y LIMITACIONES	5
1.5 SOLUCIÓN PROPUESTA	6
1.6 METODOLOGÍA.....	7
1.7 ESTRUCTURA DEL DOCUMENTO.	8
2. ANÁLISIS DE SIRA.....	10
2.1 DESCRIPCIÓN DEL SISTEMA SIRA.....	10
2.1.1 <i>Antecedentes y situación actual</i>	10
2.1.2 <i>Perspectiva del producto</i>	12
2.1.3 <i>Declaración de objetivos</i>	12
2.1.4 <i>Diagrama de contexto</i>	12
2.2 REQUERIMIENTOS ACTUALES DEL SISTEMA.....	14
2.2.1 <i>Requerimientos identificados en entrevista con el CEL LANIA</i>	14
2.2.2 <i>Requerimientos funcionales</i>	14
2.2.3 <i>Requerimientos no funcionales</i>	15
2.2.4 <i>Características de los usuarios</i>	16
2.2.5 <i>Mapa de Actores</i>	17
2.3 IDENTIFICACIÓN DE FUNCIONALIDADES DE SIRA.....	18
2.3.1 <i>Identificación de procesos clave</i>	18
2.3.2 <i>Diagrama de Casos de Uso</i>	21
2.3.3 <i>Listado de Casos de Uso</i>	22
2.4 ANÁLISIS DE DATOS.....	24
2.4.1 <i>Objetos identificados</i>	24
2.4.2 <i>Modelo de Dominio</i>	26
2.4.3 <i>Diagrama de Clases general</i>	27
2.4.4 <i>Diagrama Entidad-Relación</i>	29
3. DISEÑO DE LOS SERVICIOS IDENTIFICADOS.....	32
3.1 SIRA EN UNA ARQUITECTURA ORIENTADA A SERVICIOS.	32
3.2 ELECCIÓN DE UNA ARQUITECTURA SOA BASADA EN SERVICIOS REST.....	33
3.3 SERVICIOS IDENTIFICADOS DENTRO DEL SISTEMA SIRA.....	35
3.4 DIAGRAMA ARQUITECTÓNICO DE SIRA.....	36
3.5 DISEÑO GENERAL DE LOS SERVICIOS EN SIRA.....	38
3.6 CAPAS DE DISEÑO DE LOS SERVICIOS.....	39
3.6.1 <i>Capa de Modelado y Acceso a Datos</i>	40
3.6.2 <i>Capa de Servicio</i>	40
3.6.3 <i>Capa de Controladores REST</i>	40
3.6.4 <i>Diagrama de diseño arquitectónico de los servicios por capas</i>	41
3.7 FILTRO DE SEGURIDAD EN LOS SERVICIOS.	41

3.7.1	<i>Autenticación basada en Jason Web Token.</i>	43
3.7.2	<i>Diagrama del proceso de Autenticación de usuarios basado en JWT.</i>	45
3.7.3	<i>Diagrama del proceso de acceso a un recurso REST basado en JWT.</i>	46
3.8	SERVICIO REST DE USUARIOS.....	47
3.8.1	<i>Alcance de funcionalidades del servicio contra casos de uso.</i>	47
3.8.2	<i>Diseño de acciones y métodos del Servicio REST Usuarios.</i>	47
3.8.3	<i>Descripción de endpoints del servicio de usuarios.</i>	50
3.9	SERVICIO REST DE CATÁLOGOS	51
3.9.1	<i>Alcance de funcionalidades del servicio contra casos de uso.</i>	51
3.9.2	<i>Diseño de acciones y métodos del Servicio REST Catálogos.</i>	51
3.9.3	<i>Descripción de endpoints del Servicio de Catálogos.</i>	52
3.10	SERVICIO REST DE PARTICIPACIONES	55
3.10.1	<i>Alcance de funcionalidades contra casos de uso.</i>	55
3.10.2	<i>Diseño de acciones y métodos del Servicio REST de Participaciones.</i>	56
3.10.3	<i>Descripción de endpoints del Servicio de Participaciones.</i>	57
3.11	SERVICIO REST DE NOTIFICACIONES.....	59
3.11.1	<i>Alcance de funcionalidades contra casos de uso.</i>	59
3.11.2	<i>Diseño de acciones y métodos del servicio REST de notificaciones.</i>	60
3.11.3	<i>Descripción de endpoints del Servicio de Notificaciones.</i>	61
4.	ANÁLISIS Y DISEÑO DE LA INTERFAZ DE USUARIO RESPONSIVA.	62
4.1	ELECCIÓN DE LA TECNOLOGÍA PARA LA INTERFAZ DE USUARIO.....	62
4.2	DISEÑO Y ESTRUCTURA DE LOS COMPONENTES DE LA INTERFAZ.....	63
4.2.1	<i>Wireframes y diseño.</i>	64
4.2.2	<i>Estructura de los componentes con el framework VuetifyJS.</i>	72
5.	DESCRIPCIÓN DEL PROTOTIPO FUNCIONAL IMPLEMENTADO.	75
5.1	INSTALACIÓN Y EJECUCIÓN DEL CLIENTE WEB.....	75
5.2	INSTALACIÓN Y EJECUCIÓN DEL PROYECTO BACKEND.....	76
5.3	DESCRIPCIÓN DE FUNCIONALIDADES DEL PROTOTIPO	77
5.3.1	<i>Registro de usuario aspirante e ingreso al sistema.</i>	77
5.3.2	<i>Funcionalidades del rol aspirante.</i>	81
5.3.3	<i>Funcionalidades del rol de seguimiento.</i>	89
5.3.4	<i>Funcionalidades del rol administrador.</i>	92
5.3.5	<i>Cerrar sesión del usuario.</i>	104
6.	PRUEBAS Y CONCLUSIONES.	105
6.1	CASOS DE PRUEBA.....	105
6.1.1	<i>Casos de prueba de funcionalidades del aspirante.</i>	105
6.1.2	<i>Casos de prueba de funcionalidades del administrador.</i>	109
6.1.3	<i>Casos de prueba de funcionalidades del usuario seguimiento y administrador.</i>	114
6.2	CONCLUSIONES DEL PROYECTO.	116
6.3	TRABAJOS A FUTURO.	118
7	REFERENCIAS	119
	ANEXO A. ESPECIFICACIONES DE CASOS DE USO.	124

Índice de figuras.

Figura 1.	Diagrama de contexto de SIRA.....	13
Figura 2.	Mapa de Actores.....	17
Figura 3.	Diagrama de casos de uso.....	21
Figura 4.	Modelo de dominio	26
Figura 5.	Diagrama de clases general.....	28
Figura 6.	Diagrama Entidad-Relación.....	30
Figura 7.	Diagrama Arquitectónico de SIRA.....	37
Figura 8.	Diagrama de diseño arquitectónico por capas.....	41
Figura 9.	Proceso de autenticación de usuario basado en JWT.....	45
Figura 10.	Proceso de acceso a un recurso REST basado en JWT.....	46
Figura 11.	Modelos del Servicio Usuario.....	48
Figura 12.	Modelos y clases auxiliares del proceso de registro de usuarios.....	49
Figura 13.	Diseño de clases del servicio de catálogos.	52
Figura 14.	Diagrama de clases de Participaciones y sus relaciones.	56
Figura 15.	Diagrama de clase del servicio notificaciones para envío de correos..	60
Figura 16.	Diagrama de clase del controlador de notificaciones.....	60
Figura 17.	Pantalla de inicio de sesión.	64
Figura 18.	Pantalla de registro de usuario aspirante.....	65
Figura 19.	Wireframe de confirmación de cuenta del usuario.....	66
Figura 20.	Wireframe de layout general de la aplicación.	67
Figura 21.	Wireframe general para CRUD de catálogos.....	68
Figura 22.	Wireframe de formulario de nuevo registro para CRUD.	69
Figura 23.	Wireframe de participación del aspirante.....	70
Figura 24.	Wireframe de diseño de dashboard.....	71
Figura 25.	Diagrama de Componentes del Frontend.....	74
Figura 26.	Pantalla de registro del aspirante.....	78
Figura 27.	Pantalla de correo de confirmación de cuenta de aspirante.	79
Figura 28.	Pantalla de confirmación de cuenta exitosa.....	79
Figura 29.	Pantallas de ingreso y registro en resolución móvil.	80
Figura 30.	Opciones del menú del aspirante.....	81
Figura 31.	Pantalla de convocatorias disponibles del aspirante.....	82
Figura 32.	Pantalla de detalle de la convocatoria.	83
Figura 33.	Registro exitoso a convocatoria en resolución móvil.	84
Figura 34.	Pantalla de correo de registro a convocatoria exitoso.	85
Figura 35.	Vista "Mis Participaciones" del aspirante.	86
Figura 36.	Detalle de la participación para el usuario aspirante.....	87
Figura 37.	Detalle de participación con un requisito entregado.	88
Figura 38.	Vista de Dashboard del usuario seguimiento.....	89
Figura 39.	Pantalla de seguimiento de convocatoria	90
Figura 40.	Pantalla de seguimiento a la participación del aspirante.	91
Figura 41.	Estatus de la participación del aspirante (resolución móvil).....	92
Figura 42.	Menú desplegable del usuario administrador.....	93
Figura 43.	CRUD de Convocatorias para el rol administrador.	94
Figura 44.	CRUD de Convocatorias en resolución móvil	95
Figura 45.	Formulario de creación de nueva convocatoria.	96

Figura 46.	Botón de acción de requisitos de convocatoria.....	97
Figura 47.	Diálogo de requisitos de convocatoria.	97
Figura 48.	CRUD de programas educativos para el administrador.....	98
Figura 49.	Nuevo prog. educativo, en resolución escritorio y móvil.	99
Figura 50.	CRUD de requisitos para el administrador.....	100
Figura 51.	Agregar nuevo requisito, resolución de escritorio y móvil.	101
Figura 52.	CRUD de usuarios internos para el rol administrador.....	102
Figura 53.	CRUD de usuarios internos en resolución móvil.....	102
Figura 54.	Diálogo para nuevo usuario interno del rol admin.....	103
Figura 55.	Botón de salir o cerrar sesión.	104

Índice de Tablas

Tabla 1.	Características de los usuarios.....	16
Tabla 2.	Listado de Casos de Uso.	23
Tabla 1.	Endpoints del servicio REST usuarios.....	50
Tabla 3.	Endpoints del servicio de catálogos.	54
Tabla 4.	Descripción de endpoints del Servicio de Participaciones.....	58
Tabla 5.	CP de registro de usuario aspirante e inicio de sesión.....	106
Tabla 6.	CP de registro de participación del aspirante a una convocatoria.....	107
Tabla 7.	CP de subida de archivos de requisitos de convocatoria a la participación del aspirante.....	108
Tabla 8.	CP de inicio de sesión del administrador.....	109
Tabla 9.	CP de alta, baja, y modificaciones de programas educativos.	110
Tabla 10.	CP de alta, baja, y modificaciones de convocatorias.....	112
Tabla 11.	CP de alta, baja y modificaciones de requisitos.	113
Tabla 12.	CP de alta, baja y modificaciones de usuarios internos.	114
Tabla 13.	CP de usuario seguimiento valida archivos subidos por el aspirante....	114
Tabla 14.	CP de visualizar estatus de la convocatoria	115
Tabla 15.	CP de visualizar estadísticas del Dashboard	115

ANEXO A.

A1.	CU-01 “Crear cuenta de usuario aspirante”.....	124
A2.	CU-02 “Ingresar al sistema”.....	125
A3.	Caso de Uso CU-03 “CRUD de Programas Educativos”.....	127
A4.	Caso de Uso CU-04 “CRUD de Convocatorias”.....	128
A5.	Caso de Uso CU-05. “Consultar convocatorias disponibles”.....	129
A6.	Caso de Uso CU-06. “Registrar Participación en Convocatoria”	129
A7.	Caso de Uso CU-07 “Subir documentación en convocatoria”	130
A8.	Caso de Uso CU-08 “Verificar documentación del aspirante”	132
A9.	Caso de Uso CU-09 “Consultar estado de convocatorias del aspirante”.....	133
A10.	Caso de Uso CU-10 “Consultar estatus general de convocatorias”.....	133
A11.	Caso de Uso CU-11 “Consumir datos generales de aspirantes”.....	134
A12.	Caso de Uso CU-12 “CRUD Requisitos”	135
A13.	Caso de Uso CU-13 “Asignar requisitos a convocatorias”	136
A14.	Caso de Uso CU-14 “CRUD de Usuarios”	137

1. Introducción

Actualmente, dentro del Centro de enseñanza LANIA, se cuenta con la plataforma “SIRA” (Sistema de Registro de Aspirantes), donde los aspirantes y público en general pueden registrarse a las convocatorias de los diferentes programas educativos, eventos, cursos y talleres que se ofrecen, así como subir la documentación y dar seguimiento al proceso de admisión de los mismos.

Dicha plataforma ofrece el registro, control y seguimiento de avance a dichas convocatorias tanto del lado de los aspirantes como del personal administrativo de LANIA como herramienta para las gestiones pertinentes.

En los últimos años, en lo referente al desarrollo de aplicaciones web modernas, se tiene una tendencia clara hacia la implementación de aplicaciones que consumen o comparten información a través de servicios [7], [22], lo cual, de la mano con el desarrollo acelerado de las tecnologías en la nube, proveen al software de importantes características de disponibilidad, escalabilidad y mantenibilidad.

Por otro lado, aunado al desarrollo de las redes de comunicación y la infraestructura que conforma hoy día el Internet, se tiene una fuerte tendencia en el uso de dispositivos móviles como *smartphones* y tabletas para acceder a servicios y aplicaciones, situación que impone adoptar ciertas directrices al momento desarrollar software, las cuales van orientadas a la adecuada visualización de dichas aplicaciones en este tipo de dispositivos.

El presente trabajo pretende plasmar el desarrollo de un prototipo funcional del sistema SIRA; adoptando los dos aspectos anteriormente mencionados: la implementación de una Arquitectura Orientada a Servicios (SOA), y la implementación de una interfaz de usuario que permita una adecuada visualización en dispositivos móviles para los aspirantes al Centro de Enseñanza LANIA (CEL).

1.1 Planteamiento e identificación del problema

En la actualidad, la plataforma SIRA se encuentra desarrollada bajo un enfoque de PHP, haciendo uso del Framework *CodeIgniter* y el manejador de base de datos *PostgreSQL*. Cuenta también con una interfaz de usuario no responsive, la cual no permite una adecuada visualización con dispositivos móviles.

Si bien, la plataforma SIRA ofrece grandes ventajas tecnológicas como punto de contacto entre LANIA y los aspirantes a los programas educativos ofrecidos; se tienen dos áreas de oportunidad importantes a mejorar en relación a la arquitectura y diseño de sus componentes. Por un lado, el patrón de diseño Modelo Vista Controlador, provee de una fuerte dependencia del tipo de tecnología en la cual se encuentra actualmente desarrollada (en este caso el *Framework CodeIgniter*), y a su vez contiene cierto grado de acoplamiento tanto entre los mismos componentes de negocio, como desde éstos hacia la capa de presentación; lo cual conlleva por ejemplo, a no poder realizar de manera transparente un cambio de tecnología en la interfaz de usuario, esto sin tener que realizar modificaciones dentro de los mismos controladores PHP, donde además se encuentra implementada la lógica de negocio del sistema. Por otro lado, el diseño actual no permite una interoperabilidad entre SIRA y otros sistemas de LANIA, característica que es muy importante en miras hacia el futuro en los desarrollos de los sistemas del CEL.

Derivado de lo anterior, es de suma importancia la oportunidad de mejorar la plataforma mediante una nueva propuesta donde se pueda identificar claramente las responsabilidades de sus componentes, y a su vez les permita interactuar inclusive de manera independiente con otros sistemas o procesos que se alimenten de la información manejada por SIRA.

Aunado a los puntos señalados en el diseño arquitectónico de la aplicación, y como otra oportunidad importante de mejora, se tiene que la interfaz de usuario actual si bien es funcional, no cuenta con características de visualización responsive en clientes móviles como *smartphones* y tabletas; dispositivos utilizados por la mayoría

del público que accede desde Internet hoy en día, incidiendo de manera negativa en los usuarios al propiciar una visualización poco adecuada, situación que incluso pudiera originar cierto desinterés por parte de los aspirantes.

1.2 Objetivo general y específicos

1.2.1 Objetivo General.

Desarrollar e implementar un prototipo funcional de “SIRA”, con un enfoque orientado a servicios y una interfaz responsiva que permita una adecuada visualización para dispositivos móviles.

1.2.2 Objetivos Específicos.

- Recolectar las necesidades actuales de funcionalidad para el seguimiento al ingreso de aspirantes.
- Identificar los servicios con que debe contar la plataforma dadas las necesidades recolectadas.
- Realizar una propuesta de interfaz de usuario responsiva que brinde una adecuada visualización para móviles.
- Programar un prototipo que implemente los servicios y funcione con la interfaz propuesta.

1.3 Justificación

En cuanto a las preferencias actuales en el desarrollo de productos de software, en los últimos años existe una fuerte tendencia en el desarrollo enfocado a la arquitectura de servicios y micro servicios, puesto que se provee al producto de importantes características, entre las cuales destacan: un software altamente mantenable en el tiempo, permitir ciclos de desarrollo incrementales, crecimiento o escalabilidad horizontal, bajo acoplamiento de componentes, orientación a mensajes, entre otras [1][7].

Hoy en día, las arquitecturas de tipo SOA han demostrado gran eficacia en el proceso de reducir el grado de acoplamiento entre los procesos de negocio del software, permitiendo la independencia tecnológica de sus componentes, mediante el uso de interfaces o contratos que se exponen como servicios y se caracterizan por ser reutilizables y proveer interoperabilidad con otras aplicaciones [19].

Por otro lado, según lo reportado por “*Digital 2021*” a enero de 2021, el 95.9% del total de usuarios de Internet en México acceden a través dispositivos móviles, y es de notar que, de acuerdo al tipo de tráfico consumido durante diciembre 2020, específicamente el correspondiente a páginas *WEB*, es del 57.9% a través de este tipo de dispositivos, mientras que el 42% fue accedido a través de laptops y computadoras de escritorio [2]. Teniendo tal escenario, es imprescindible dar importancia en mejorar el servicio a usuarios que consumen páginas web mediante el uso de dispositivos móviles.

En relación a la importancia de una interacción positiva del usuario con un producto de software, es de notar que “*...un buen diseño de interfaz puede permitirle al usuario mantener una interacción más simple y eficiente con el producto web...*”, “*...y si, además, el producto ofrece propiedades de Diseño Web Responsivo (RWD), es mucho más probable que los visitantes desde dispositivos móviles experimenten una interacción más positiva que los incite a visitar y utilizar el producto nuevamente...*” [3].

Aunado a lo anterior, la identificación de las necesidades actuales en el proceso del ingreso de aspirantes a los programas educativos que ofrece LANIA, es de gran utilidad para poder incorporar nuevas características y mejoras, las cuales han sido identificadas a lo largo del tiempo con la experiencia del uso del sistema en su versión actual, y al mismo tiempo impactar positivamente hacia un mejor manejo y seguimiento de los aspirantes a los programas educativos del CEL.

1.4 Alcances y limitaciones

El proyecto “SIRA orientada a servicios”, tiene los siguientes alcances:

- La identificación de necesidades actuales de funcionalidades, derivadas del proceso de admisión de estudiantes a las distintas convocatorias que ofrece LANIA, donde se puntualizan las funcionalidades actuales de la plataforma, así como los nuevos requerimientos.
- La descripción de cada servicio requerido de acuerdo al análisis y diseño detallado de las funcionalidades identificadas, implementando una arquitectura orientada a servicios (SOA).
- El diseño de una interfaz de usuario que se adapte a las funcionalidades identificadas y analizadas, y que al mismo tiempo contemple cualidades responsivas que proporcionen una adecuada visualización en dispositivos móviles.
- La implementación de un prototipo funcional, cuya finalidad será reflejar de manera integral tanto la funcionalidad de los servicios identificados, como la interfaz de usuario propuesta, no así con el objetivo de implementar una versión final y productiva de la plataforma.
- Se contempla la entrega del código fuente en medios digitales de dicho prototipo para los fines que convengan a la institución.
- La finalidad del proyecto es realizar una nueva propuesta que dé solución a los requerimientos actuales de SIRA, y no precisamente un documento de comparación exhaustiva con la versión actual del sistema.

1.5 Solución propuesta.

Tomando en cuenta lo expuesto anteriormente, se propone desarrollar e implementar un prototipo funcional del sistema “SIRA”, implementando una arquitectura de diseño de tipo Orientada a Servicios (SOA), permitiendo reducir el grado de acoplamiento en la interacción de los componentes de la lógica de negocio que será disgregada en dichos servicios, lo que permitirá obtener una mayor independencia y modularidad entre las responsabilidades de los mismos, e inclusive, ofrecer el beneficio de comunicarse de manera independiente con otros sistemas o herramientas con las que ya cuente el CEL (Centro de Enseñanza LANIA), y que impactan directa o indirectamente en el proceso de admisión de aspirantes.

Así mismo, se propone la implementación de una nueva interfaz de usuario mediante el uso del Framework Vuetify.js, el cual además de seguir las directrices del marco de diseño *Material Design de Google*, proporciona a sus componentes de características responsivas para dispositivos móviles, lo que provee de una visualización adecuada al usuario en dichos dispositivos, facilita el desarrollo de la propuesta, y al mismo tiempo da la oportunidad de implementar nuevos requerimientos visuales del sistema.

Para materializar dicha solución se propone la implementación de un prototipo funcional que integra ambos aspectos mencionados, y que será vital como punto de partida para el desarrollo futuro de una versión final y estable de la plataforma SIRA.

1.6 Metodología

Para la ejecución del proyecto se siguieron las siguientes etapas:

- ✓ Fase 1: Análisis y Diseño general:

1). Descripción de manera general y marco contextual, antecedentes y situación actual del sistema “SIRA”.

2). Realización de “Especificación de Requerimientos de Software”, (basándose en el estándar IEEE 830), describiendo el análisis de requerimientos funcionales y no funcionales de las necesidades actuales en la plataforma, relacionadas al proceso de admisión de aspirantes, apoyado de la metodología UML (*Unified Model Language*) para la integración de diagramas y casos de uso.

- ✓ Fase 2: Iterativa por cada servicio identificado:

3). Descripción de cada uno de los servicios identificados en la lógica de negocio, apoyándose de diagramas de clases y modelado de datos, descripción de componentes. En este punto, se trabajó de manera paralela el desarrollo del prototipo, dividido por entregas parciales de cada uno de los servicios implementados. Para desarrollo backend de dicho prototipo, se utilizó el Framework “*SpringBoot*” en su versión 2.4.4, en el lenguaje *Java openJDK* en su versión 14. Para la base de datos se utilizó *PostgreSQL* en su versión 13.

4) Diseño y descripción de componentes de la interfaz web. Al igual que el punto anterior, esta etapa se trabajó paralelamente a la implementación *frontend* del prototipo, generando avances de manera iterativa. Para el desarrollo de la interfaz se utilizó el Framework *Vuetify* en su versión 2.4, basado en el Framework *Vue.js* del lenguaje *JavaScript*. Es de importancia remarcar que dicho *framework* sigue las directrices del marco de diseño *Material Design de Google*, y que a su vez provee de características responsivas para dispositivos móviles en todos sus componentes.

- ✓ Fase 3: Pruebas funcionales.

5) Finalmente, se contempló una etapa de pruebas funcionales tanto de la parte de *Backend* como del *Frontend* del prototipo programado. Donde se describirán los resultados obtenidos y trabajos a futuro.

1.7 Estructura del documento.

El presente documento de trabajo recepcional, está compuesto por seis capítulos incluyendo este capítulo de introducción. A continuación, se describe cada uno de ellos:

En el capítulo dos se presenta la fase de análisis, donde se plasma la primera y segunda fase de la metodología seguida, “Especificación de Requerimientos de Software”. En este capítulo se identifican antecedentes y situación actual, perspectiva del producto, declaración de objetivos, diagrama de contexto, requerimientos funcionales y no funcionales, características de los usuarios, identificación de procesos de SIRA, objetos identificados, modelo de dominio, diagrama de clases, entre otros. Adicional y como resultado del trabajo de análisis de esta fase, se incluyen todas las especificaciones de casos de uso como ANEXO A del presente documento.

En el capítulo tres se aborda lo relacionado al diseño de los servicios de SIRA, se describe cada uno de los servicios identificados, el diagrama arquitectónico de la solución, el diseño general de los servicios, sus capas de diseño y la capa de seguridad.

En el capítulo cuatro, se describe el análisis y diseño de la interfaz responsiva. Se exponen las razones de la tecnología utilizada, algunos *wireframes* de diseño, así como la estructura y descripción de los componentes de la interfaz de usuario.

En el capítulo cinco se describe el prototipo funcional implementado, la instalación o ejecución, aspectos generales de configuración y se describen las funcionalidades por cada tipo de usuario mostrando las pantallas correspondientes.

Finalmente, en el capítulo seis, se presentan los casos de prueba realizados a al prototipo funcional desarrollado, así como los resultados y conclusiones obtenidas en base a los objetivos planteados y trabajos a futuro.

2. Análisis de SIRA

En el presente capítulo se describen cada una de las etapas que se desarrollaron como parte del análisis de requerimientos del sistema SIRA, para las cuales se tomó como referencia el documento “Especificaciones de Requisitos de Software (ERS) del IEEE 830-1998” [18]. Dichas etapas fungieron como base para posteriormente poder diseñar el sistema, permitiendo así dimensionar en primera instancia la perspectiva del producto de software, su entorno e interacción con otros sistemas, la identificación puntual de los requerimientos funcionales y no funcionales, los casos de uso y actores involucrados, y a partir de ahí, el proceso de análisis de datos o dominio del sistema, identificando los objetos y sus características, y la abstracción de las clases-entidad, conceptos que desembocan en el diseño del diagrama de clases y el diagrama entidad relación descritos en los últimos puntos del capítulo.

2.1 Descripción del sistema SIRA.

2.1.1 Antecedentes y situación actual.

Dentro de las herramientas tecnológicas con las que cuenta el centro de enseñanza LANIA, se encuentra el sistema SIRA (Sistema de Registro de Aspirantes), como un punto de contacto y control dentro del proceso de admisión de aspirantes a los distintos programas educativos ofrecidos por la institución.

Actualmente, los aspirantes pueden ingresar a través de Internet por medio de un navegador web a dicha plataforma, en donde se registran con su correo electrónico, contraseña, número de teléfono y WhatsApp. Una vez registrado, se presenta la pantalla principal donde aparecen las participaciones a las convocatorias en las que se encuentra participando.

El usuario puede consultar las convocatorias disponibles, la descripción y detalles de las mismas, así como los requisitos que debe cumplir para participar en alguna de ellas. Al mismo tiempo, le permite registrarse en las convocatorias deseadas

para comenzar a subir los documentos solicitados, dando seguimiento a la completitud de los mismos y el estatus en el que se encuentra su participación.

Por otro lado, el personal administrativo tiene acceso a la actualización y alta de los distintos programas académicos, convocatorias y documentos o requisitos correspondientes a cada una de las convocatorias. Sin embargo, actualmente no se cuenta con un tablero ejecutivo donde se pueda visualizar el estatus que guardan cada una de las convocatorias, lo que repercute en el adecuado seguimiento de las mismas.

Si bien el aspirante puede ingresar a consultar sus participaciones, es un tanto difícil localizar de manera rápida el estatus de las mismas, así mismo, no existe un canal o mecanismo de interacción con la institución de manera directa con la plataforma, como podría ser una mesa de ayuda o el uso de un API de terceros como *Whatsapp Business*, que enriquezca en mayor medida la comunicación entre los aspirantes y la institución.

Actualmente, dicha plataforma se encuentra desarrollada bajo una arquitectura cliente-servidor, en donde sus componentes se encuentran estructurados de manera monolítica, haciendo uso del Framework CodeIgniter y el manejador de base de datos Postgresql. Lo anterior, genera ciertas limitantes al momento de desarrollar nuevas funcionalidades, puesto que mantiene acoplados todos los componentes de la lógica de negocio, lo que impacta de manera negativa en su mantenimiento y actualización. De igual manera, el tener todo el proceso como una sola unidad, dificulta el intercambio de mensajes entre procesos individuales hacia otras plataformas o servicios de la misma institución, los cuales pudieran verse beneficiados por el proceso del control de SIRA y las convocatorias ofrecidas.

Aunado a lo anterior, actualmente se cuenta con una interfaz de usuario no responsive, la cual no permite una adecuada visualización de los elementos a los usuarios que ingresan mediante dispositivos móviles, como *smartphones*, tabletas y diversos gadgets que en la actualidad son comúnmente utilizados para acceder a Internet, impactando negativamente en el interés de los aspirantes.

2.1.2 Perspectiva del producto.

Partiendo de la publicación de las convocatorias a los diferentes programas educativos ofrecidos por LANIA, y como punto de enlace entre los interesados y el centro de enseñanza, es imperante contar con una plataforma que permita el registro y la administración de la información de dichos candidatos, la cual provea de una interfaz de usuario responsiva para dispositivos móviles y que permita, tanto al aspirante como al personal de seguimiento del CEL, conocer en todo momento el estatus de las participaciones a las convocatorias ofrecidas.

Se pretende que dicha plataforma sirva como repositorio de información de los datos generales de los aspirantes, así como de su documentación, y que, de convertirse en un futuro como estudiantes, esta información pueda ser consumida a través de un API por el Sistema de Control Escolar (SICEL), el cual está relacionado con la gestión de actividades enfocadas al proceso académico – administrativo de LANIA.

2.1.3 Declaración de objetivos.

Se pretende desarrollar un prototipo funcional estructurando sus componentes en una Arquitectura Orientada a Servicios, respondiendo a las necesidades actuales del proceso de aspirantes a las convocatorias de LANIA, las cuales se detallarán puntualmente en la siguiente sección. Dicho prototipo contará con una interfaz que será accedida a través de un navegador web, y tendrá características responsivas que permitirán una visualización adecuada a usuarios que ingresen mediante dispositivos móviles.

2.1.4 Diagrama de contexto.

La figura 1 muestra el diagrama de contexto de SIRA, en donde se pueden apreciar los distintos entes, actores y sistemas que interactúan con la plataforma.

Como se puede apreciar en el diagrama, se tiene por un lado el público interesado en participar en las convocatorias ofrecidas por el CEL, y por el otro lado el

personal administrativo y demás sistemas de LANIA que realizan el seguimiento y obtienen información de las participaciones.

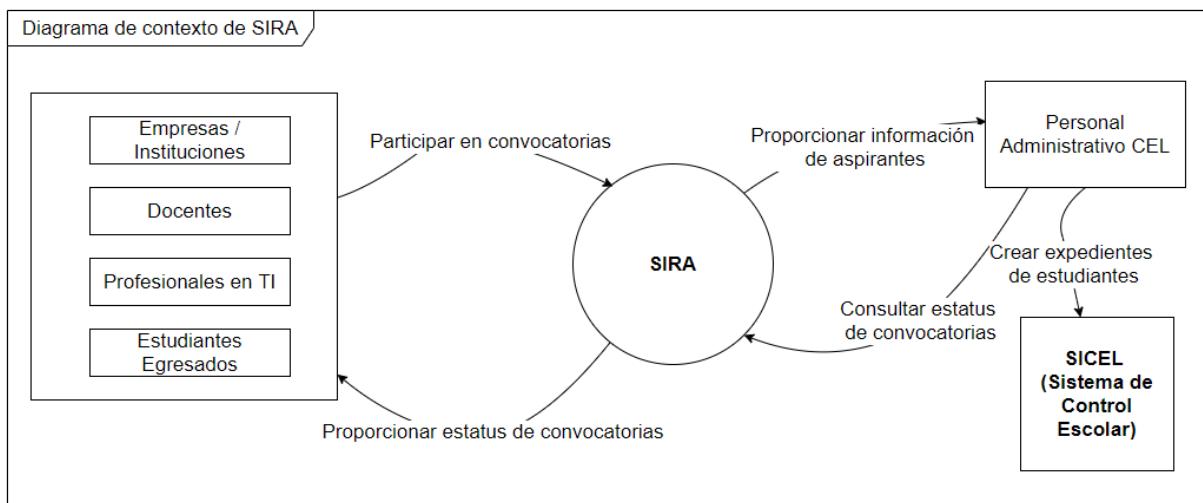


Figura 1. Diagrama de contexto de SIRA.

2.2 Requerimientos actuales del sistema.

2.2.1 Requerimientos identificados en entrevista con el CEL LANIA.

El día 6 de mayo de 2021, se llevó a cabo una reunión para obtener información valiosa respecto al análisis de requerimientos del sistema SIRA, en donde se entrevistó al personal del Centro de Enseñanza LANIA, y en la cual se pudieron identificar las necesidades actuales del sistema SIRA. Dicha reunión fue de gran ayuda como parte del análisis de los requerimientos del sistema, los cuales se describen en los siguientes dos apartados como “Requerimientos Funcionales y No Funcionales”.

2.2.2 Requerimientos funcionales.

Dentro de los requerimientos funcionales se identificaron los siguientes:

- La administración de usuarios del sistema (altas, bajas, cambios).
- La administración de programas académicos (altas, bajas, modificaciones).
- La administración de convocatorias (altas, bajas, modificaciones).
- La administración de requisitos de convocatorias (altas, bajas, modificaciones), y poder vincular los requisitos correspondientes a cada convocatoria.
- El seguimiento de convocatorias mediante un tablero ejecutivo por estatus y aspirantes.
- Poder visualizar los aspirantes por convocatoria.
- Poder visualizar el número total de aspirantes actuales.
- La evaluación de cumplimiento de requisitos por aspirante y por convocatoria.
- Asignar un estatus del aspirante dentro de la convocatoria.
- La posibilidad de brindar una herramienta de soporte al aspirante durante su proceso, (herramienta de terceros como mesa de ayuda o *Whatsapp Business*).
- El registro de datos generales del aspirante.

- Permitir visualizar al aspirante las distintas convocatorias.
- Funcionalidad para generar participación del aspirante en una convocatoria.
- Funcionalidad para subir la documentación del aspirante en las convocatorias donde participa.
- Permitir al aspirante visualizar el estatus de su participación.
- Permitir al aspirante recibir notificaciones (correos electrónicos).
- Generar un servicio web o API para exportar o consumir datos generales y documentación de los aspirantes.

2.2.3 Requerimientos no funcionales.

Dentro de los requerimientos no funcionales se detectaron los siguientes:

- El desarrollo de la interfaz del prototipo deberá ser de tipo web, para ser utilizado mediante un navegador. Firefox, Safari, Chrome.
- El desarrollo de una interfaz responsive que permita la navegación adecuada en dispositivos móviles como *smartphones* y tabletas.
- Se requiere el desarrollo del sistema bajo una arquitectura orientada a servicios, para poder lograr el desacoplamiento de componentes.
- Se requiere un modelado de la arquitectura del sistema, así como documentación de análisis y diseño.
- La comunicación con otras aplicaciones se deberá realizar mediante API REST, para consumo en formato JSON o XML.
- Envío de correo electrónico a través de la herramienta: MailTrap.
- Interfaz de usuario que permita indicar de manera clara al aspirante el proceso y en qué estatus se encuentra.
- Mostrar un acuerdo de confidencialidad al momento de registrarse en la plataforma.

2.2.4 Características de los usuarios.

Como parte del proceso de entrevista con el personal del CEL, se detectó la necesidad de crear dentro del sistema, identificando los perfiles de usuario descritos en la Tabla 1:

Tipo de Usuario	Descripción
Administrador	Usuario que debe tener conocimiento de la gestión de la aplicación, debería recibir una capacitación acerca del uso del Sistema. El administrador será el responsable de dar de alta, modificar y eliminar catálogos del Sistema tales como: usuarios, convocatorias, programas académicos, requisitos de las convocatorias y dar seguimiento de las mismas.
Aspirante	Persona que va a interactuar continuamente con la aplicación, y no debe tener una capacitación. Los aspirantes deberán poder darse de alta en la plataforma, registrarse en las convocatorias deseadas, así como subir la documentación requerida por la convocatoria a la cual participan. También podrán ver el seguimiento y evaluación de su participación.
Seguimiento	Usuario por parte del CEL, el cual debería recibir una capacitación del uso del sistema. Este usuario interactúa con tareas de seguimiento como la visualización de seguimiento a convocatorias, aspirantes inscritos. Sin embargo, no realizará tareas de modificación de catálogos programas educativos y convocatorias ni usuarios.

Tabla 1. Características de los usuarios.

2.2.5 Mapa de Actores.

Como diagrama de referencia se realizó el siguiente “Mapa de Actores”, para visualizar de mejor manera los actores que intervienen en la operatividad del sistema, así como la relación que tienen entre ellos, ver figura 2:

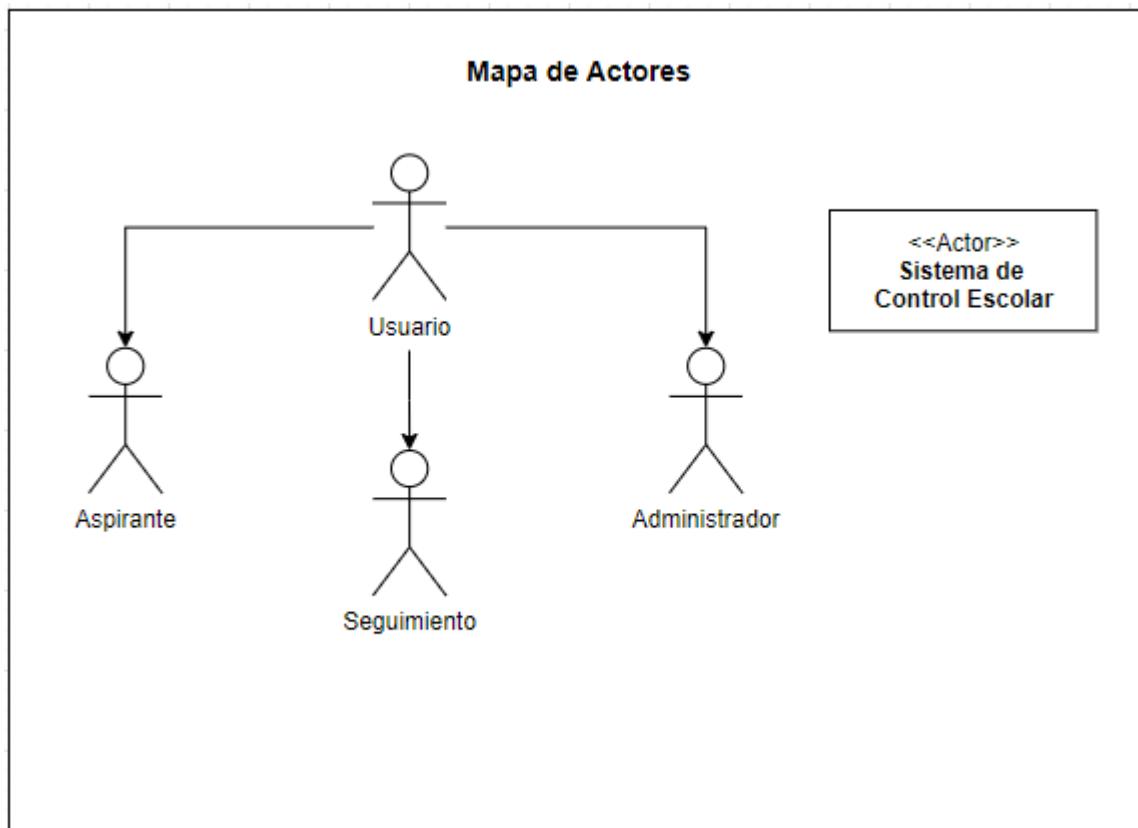


Figura 2. Mapa de Actores.

Como actores por parte de la institución se pueden observar dos actores internos del sistema: el usuario seguimiento, y el usuario administrador; así como el actor externo el Sistema de Control Escolar, que se pretende pueda consumir información de los aspirantes mediante los servicios que proporcione SIRA. Finalmente, existe el actor aspirante que será el público interesado en participar en las convocatorias.

2.3 Identificación de funcionalidades de SIRA.

2.3.1 Identificación de procesos clave.

Derivado del análisis y entrevistas con el personal de LANIA, se identificaron los siguientes procesos clave del proceso que abarca SIRA:

- **Proceso de registro, edición y eliminación de usuarios:** Proceso mediante el cual se permite al usuario administrador, registrar, modificar y eliminar usuarios de la plataforma SIRA. Este proceso debe permitir registrar usuarios aspirantes mediante datos mínimos como: nombre, apellido, número de *Whatsapp*, correo y contraseña; y también registrar usuarios correspondientes al Centro de Enseñanza LANIA, con datos mínimos como: nombre, apellido, departamento, correo y contraseña.
- **Proceso de registro, edición y eliminación de programas educativos:** Proceso mediante el cual se le permite al usuario administrador registrar, modificar y eliminar programas educativos dentro del sistema. Para poder registrar un programa educativo, se debe tomar en cuenta los datos mínimos como: nombre, clave, descripción y vigencia.
- **Proceso de registro, edición y eliminación de convocatorias:** Proceso mediante el cual se le permite al usuario administrador, registrar modificar y eliminar convocatorias al usuario administrador dentro del sistema. Para poder registrar una convocatoria se deben tomar en cuenta datos mínimos como: nombre, descripción, fecha de inicio, fecha de término, el programa educativo al que pertenece y el cupo de aspirantes disponible.
- **Proceso de consulta de convocatorias disponibles para el aspirante:** Este proceso permitirá a los aspirantes poder consultar las convocatorias disponibles dentro de la plataforma, así como los requisitos y detalles de cada una ellas. Dentro de los detalles a consultar de cada convocatoria deberán

mostrarse: la fecha de inicio, la fecha de término, la fecha de examen (si aplica) y la lista de requisitos que aplican.

- **Proceso de inscripción en convocatorias por el aspirante:** Proceso mediante el cual el aspirante puede consultar las distintas convocatorias disponibles y a su vez registrar su participación en las mismas. En este proceso se le debe indicar al aspirante de manera clara los pasos a seguir para su correcto registro de la participación, así mismo se le deberá notificar vía correo el registro de su participación una vez finalizado.
- **Proceso de entrega de documentos de requisitos de convocatorias por el aspirante:** Este proceso le permitirá al aspirante registrar cada uno de los documentos necesarios como requisito a la convocatoria en la cual está participando, y se le permitirá subirlos a la plataforma mediante archivo en formato PDF. Una vez entregado el documento se deberá indicar el estatus de entregado.
- **Proceso de consulta de estatus de convocatorias por el administrador y seguimiento.** Este proceso permitirá al usuario administrador o seguimiento, poder consultar mediante un tablero ejecutivo el número de aspirantes inscritos en una convocatoria, las convocatorias disponibles actuales, así como el estatus de las participaciones de cada convocatoria.
- **Proceso de consulta de estatus de convocatorias por el aspirante:** Este proceso debe permitir al aspirante consultar el estatus de las convocatorias en las que se encuentra participando, señalando el número de requisitos necesarios para completar su participación, así como los detalles de fechas importantes de la convocatoria.

- **Proceso de notificaciones al aspirante vía correo.** Este proceso se encargará de realizar las notificaciones correspondientes a los eventos importantes a los aspirantes vía correo. Dentro de los eventos importantes a notificar se incluyen: el mensaje con la liga para confirmar el registro de cuenta del aspirante, el correcto registro de su participación en alguna convocatoria, y un recordatorio para completar los requisitos de alguna participación, en caso de qué no los haya completado y el periodo de entrega esté por concluir.

2.3.2 Diagrama de Casos de Uso.

De acuerdo a los procesos identificados, se generaron distintos casos de uso los cuales se plasmaron de manera gráfica en el siguiente diagrama (en la siguiente sección se presentan a manera de listado).

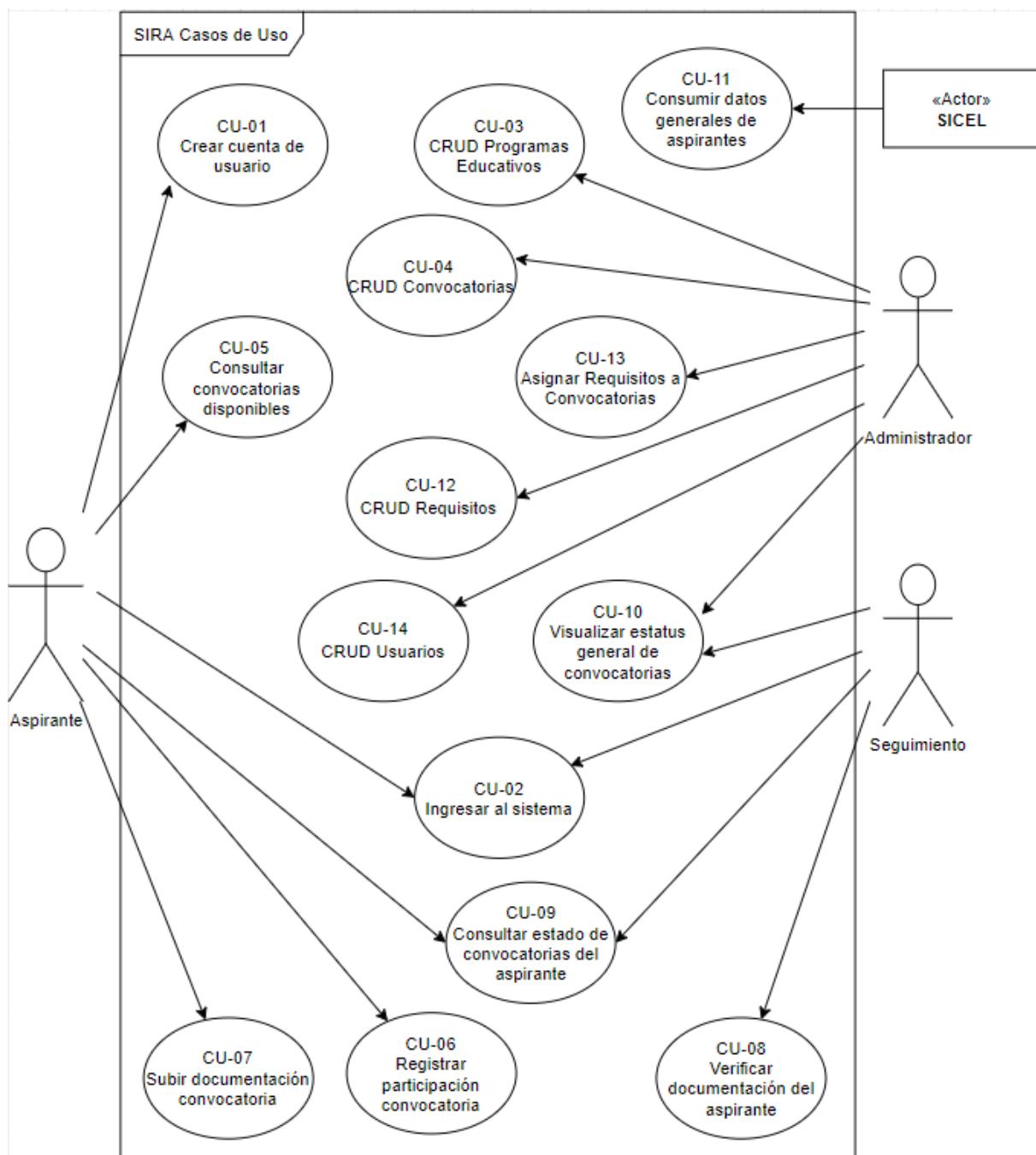


Figura 3. Diagrama de casos de uso.

2.3.3 Listado de Casos de Uso.

A continuación, se describen de manera general los casos de uso identificados, para una referencia a detalle de cada Caso de Uso, consultar el anexo A, "Especificaciones de Casos de Uso".

Clave	Requerimiento	Descripción
CU-01	Crear cuenta de usuario aspirante	El aspirante ingresa sus datos generales para crear su cuenta y recibe correo con su cuenta de usuario.
CU-02	Ingresar al sistema	El usuario ingresa al sistema mediante sus credenciales.
CU-03	CRUD Programas Educativos	Usuario Administrador captura, edita o elimina programas educativos.
CU-04	CRUD Convocatorias	Usuario Administrador captura, edita o elimina convocatorias.
CU-05	Consultar convocatorias disponibles	El aspirante consulta las convocatorias disponibles.
CU-06	Registrar participación en convocatoria	El aspirante registra su participación en una convocatoria disponible.
CU-07	Subir documentación en convocatoria	El aspirante registra su documentación en la convocatoria donde participa

CU-08	Verificar documentación del aspirante	El administrador verifica la documentación del aspirante en la convocatoria.
CU-09	Consultar estado de convocatorias del aspirante	El aspirante consulta el estatus de su participación en las convocatorias.
CU-10	Visualizar estatus general de convocatorias	El usuario seguimiento o administrador consulta estatus general de las convocatorias.
CU-11	Consumir datos generales de aspirantes	SICEL consume datos generales de los aspirantes.
CU-12	CRUD Requisitos	Usuario Administrador captura, edita o elimina requisitos.
CU-13	Asignar requisitos a convocatorias	El usuario administrador asigna los requisitos a las convocatorias.
CU-14	CRUD de Usuarios	El usuario administrador crea, modifica o elimina usuarios internos de LANIA, con rol de Seguimiento o Administrador.

Tabla 2. Listado de Casos de Uso.

2.4 Análisis de datos.

Para la realización del análisis de datos se siguieron las siguientes etapas:

- 1) Identificación de los objetos del dominio,
- 2) Generación del modelo de dominio,
- 3) Generación de diagrama de clases general y
- 4) Generación de diagrama entidad-relación.

En las siguientes subsecciones se describen los resultados de cada etapa.

2.4.1 Objetos identificados.

En este apartado se describen los objetos identificados en el análisis de datos, resultado de las actividades de entrevistas y reuniones con el personal del centro de enseñanza. A partir de este listado, se procedió a realizar el modelo de dominio y diagrama de clases.

- **Aspirantes:** Cualquier persona del público en general interesada en cursar alguno de los programas educativos ofrecidos por el centro de enseñanza LANIA.
- **Programas Académicos o Educativos:** Instrumento curricular o documento donde se especifica o detalla el proceso pedagógico, las actividades de enseñanza-aprendizaje y contenidos a desarrollar, objetivos a lograr por parte del docente, las conductas esperadas de los alumnos, así como las estrategias y recursos a emplear con este fin. Los Programas Educativos cuentan con una clave y una vigencia.
- **Convocatorias:** Es el documento formal elaborado por LANIA, mediante el cual se invita o convoca a las personas para concursar, inscribirse o participar en los distintos programas educativos ofrecidos por el CEL. Dicho documento escrito contiene de manera precisa las bases de la convocatoria. Cada convocatoria pertenece a un Programa Educativo. Un aspirante puede participar en muchas convocatorias.

- **Requisitos de convocatorias:** Condiciones que el aspirante debe cumplir para poder completar su participación en una convocatoria, pueden ser documentos de identidad o probatorios de grados de estudios, así como exámenes necesarios para participar en la convocatoria.
- **Usuarios:** Cada uno de los entes actores que acceden e interaccionan con el sistema SIRA.
- **Documento requisito:** Documento oficial que puede validar tanto la identidad del aspirante como los estudios realizados y que se pide como condición para poder participar en alguna convocatoria ofrecida por LANIA. Algunos documentos son cambiantes en el tiempo, es decir, deben ser actualizables como por ejemplo una carta de recomendación, y otros no cambian como por ejemplo el acta de nacimiento o un certificado de estudios.
- **Notificaciones:** Mensajes o alertas que informan a los usuarios sobre eventos importantes dentro del flujo de proceso de LANIA, las cuales serán básicamente correos electrónicos.
- **Participaciones:** Es el proceso individual de cada aspirante que se ha inscrito en alguna de las convocatorias correspondientes a los Programas Educativos ofrecidos por LANIA, en donde el aspirante se encuentra sujeto a un proceso de selección y deberá cumplir con cada uno de los requisitos solicitados en dicha convocatoria. Dicha participación será evaluada por el personal del CEL y podrá determinar la aceptación de la inscripción del interesado en cursar alguno de los Programas Educativos.

2.4.2 Modelo de Dominio.

En el siguiente Modelo de Dominio, se puede apreciar los objetos identificados dentro del sistema, así como la relación que tienen entre sí.

Se puede apreciar que el usuario puede ser de tipo Administrador, Aspirante o Seguimiento. El usuario Aspirante tiene participaciones y éstas pertenecen a alguna convocatoria. Dicha convocatoria a su vez pertenece a un programa académico y tiene requisitos a cumplir por parte del Aspirante mediante su participación.

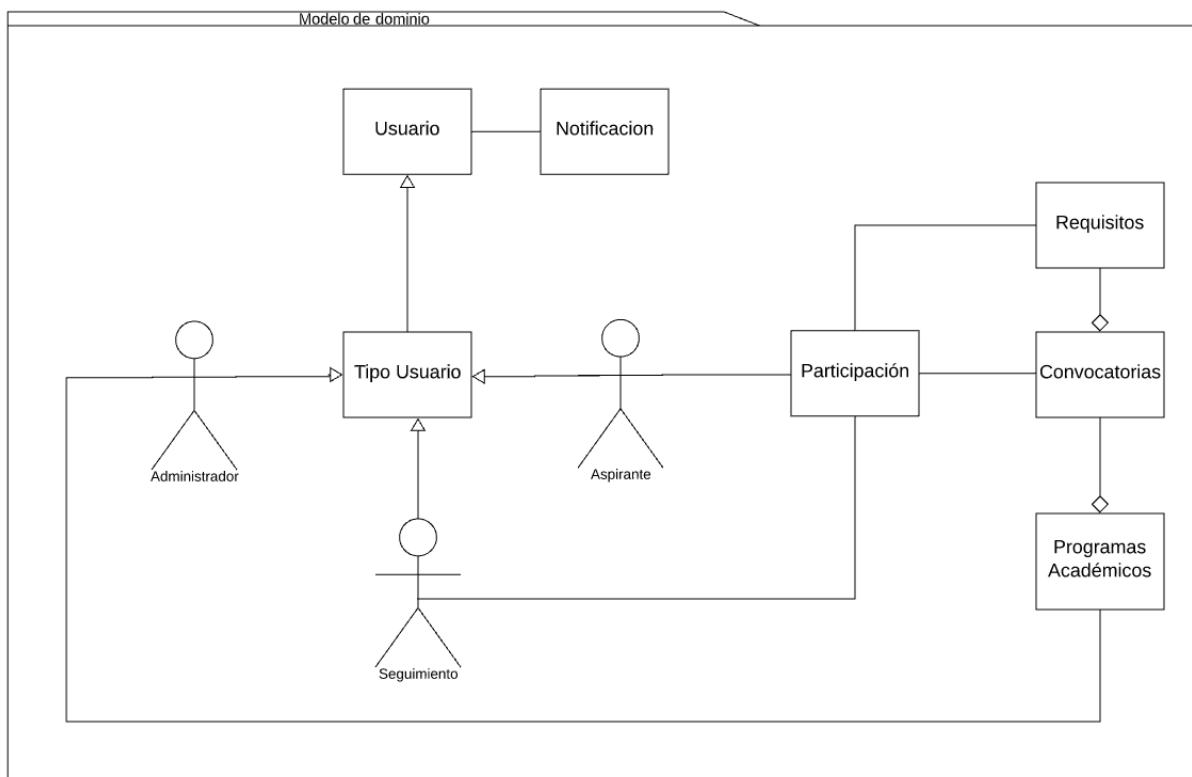


Figura 4. *Modelo de dominio*

2.4.3 Diagrama de Clases general.

De acuerdo a los objetos identificados y como resultado del análisis de datos, se elaboró un diagrama general de clases para el sistema SIRA, en donde se identificaron las siguientes clases de tipo entidad:

- **Usuario:** Clase que encapsula al usuario de manera general.
- **Aspirante:** Clase que hereda de la clase usuario y encapsula el tipo Aspirante.
- **Empleado:** Clase que hereda de la clase usuario y encapsula el tipo Empleado.
- **Participación:** Clase que encapsula la participación del aspirante.
- **Convocatoria:** Clase que encapsula la convocatoria ofrecida por LANIA.
- **ProgramaEducativo:** Encapsula el programa educativo al que pertenece la convocatoria.
- **Requisito:** Encapsula el requisito que debe cumplir el aspirante.
- **RequisitoConvocatoria:** Encapsula la relación a muchos del requisito con su convocatoria.
- **ParticipacionRequisitoConvocatoria:** Encapsula la relación a muchos y define el requisito de la participación a la convocatoria por parte del aspirante.

Cada una de las clases anteriormente listadas se describe de manera detallada en el capítulo 3, donde se plasma su interacción entre sí dentro del apartado de cada uno de los servicios descritos de SIRA.

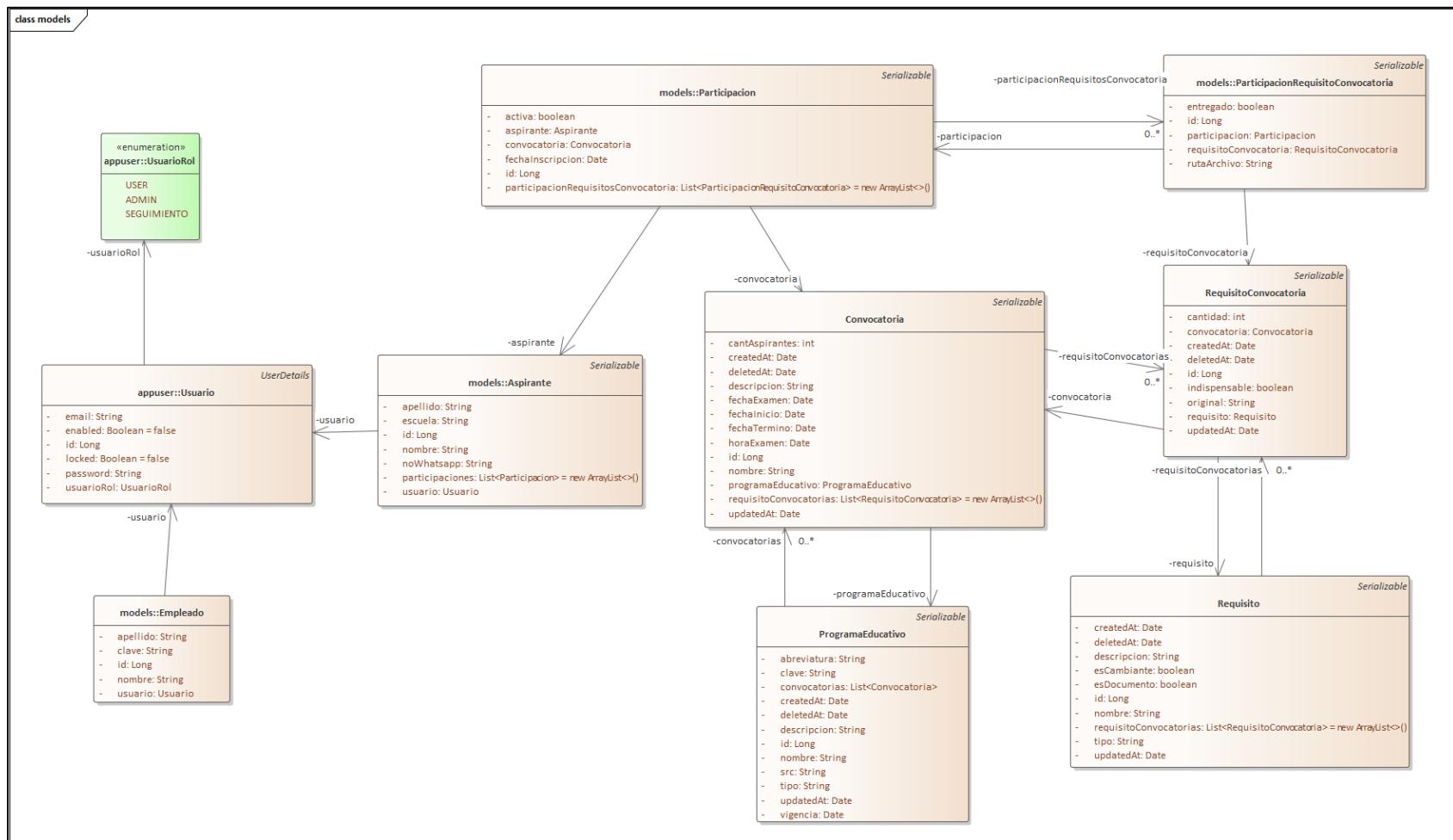


Figura 5. Diagrama de clases general.

2.4.4 Diagrama Entidad-Relación

Como resultado final del análisis de datos, se generó el diagrama Entidad-Relación en donde se observa claramente los atributos de cada una de las entidades, así como las entidades auxiliares propuestas.

Cabe resaltar que todo el proceso nace de la relación “uno a muchos” entre la entidad Aspirantes y la entidad Participaciones, donde un aspirante puede tener muchas participaciones y a su vez, ésta última es la entidad auxiliar para representar la relación “muchos a muchos” entre Aspirantes y Convocatorias.

Al realizar el análisis de relación entre entidades, se identificó la necesidad de establecer relaciones “muchos a muchos”, lo cual estableció cierto grado de complejidad en el diseño del modelo entidad relación, y donde se observó la necesidad de crear entidades auxiliares para representar dichas relaciones. Como ejemplo, se puede observar la relación entre la entidad Requisito y la entidad Convocatoria, en donde se expresa su relación “muchos requisitos pertenecen a muchas convocatorias”, mediante la entidad “*requisitos_convocatorias*”.

De igual manera, partiendo de la relación arriba descrita, para poder registrar el avance del aspirante en cada una de sus participaciones respecto al estatus de cada requisito de convocatoria, se generó la entidad auxiliar o intermedia “*participaciones_requisitos_convocatorias*”, en donde se expresa la relación “muchas participaciones tienen muchos *requisitos_convocatorias*”.

En la página siguiente se muestra el diagrama Entidad-Relación de SIRA.

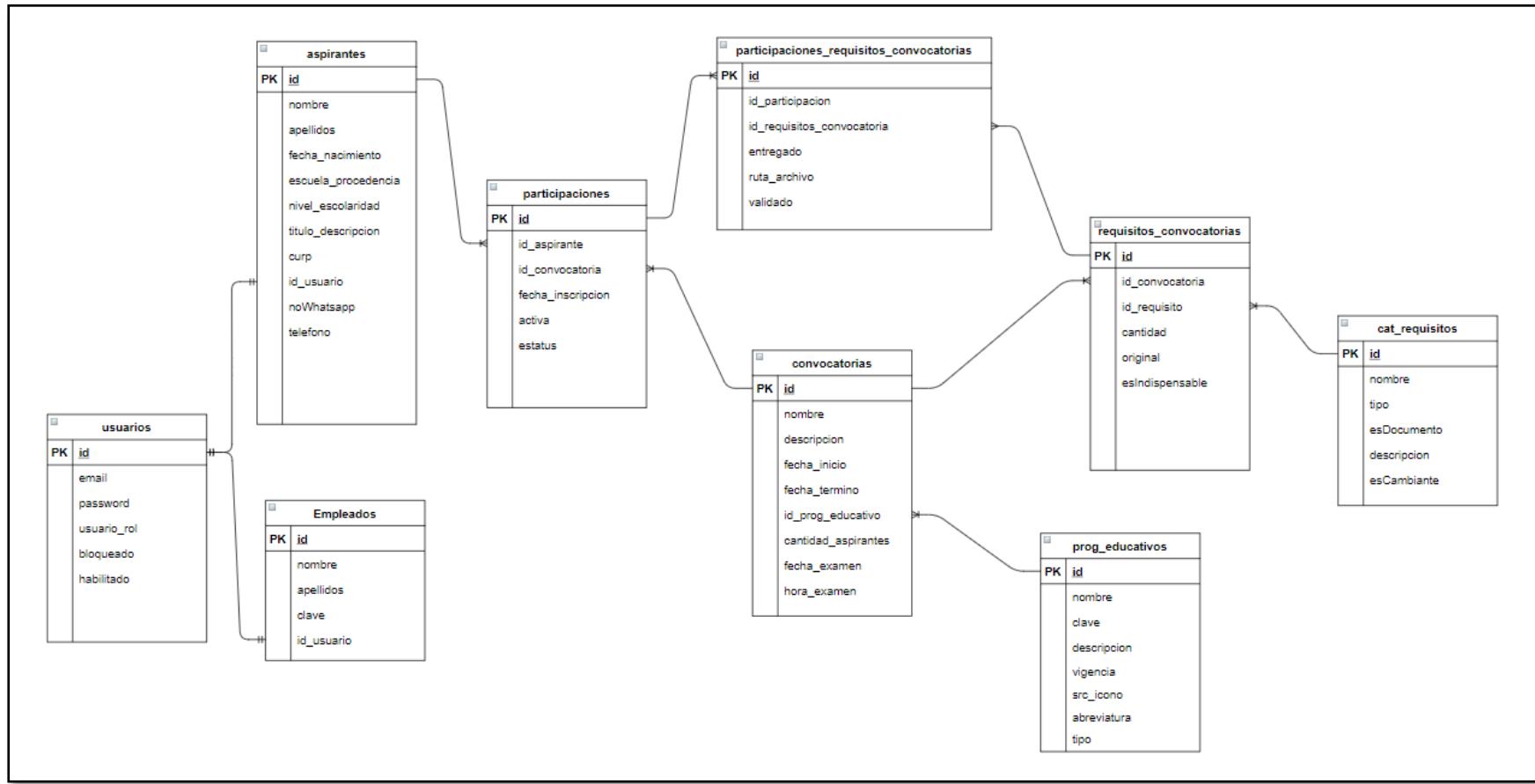


Figura 6. Diagrama Entidad-Relación.

3. Diseño de los servicios identificados.

En este tercer capítulo, se describe de manera general la elección de la arquitectura SOA, el enfoque implementado dentro del sistema SIRA, la descripción de cada uno de los servicios identificados, así como la capa de seguridad propuesta.

3.1 SIRA en una arquitectura orientada a servicios.

La Arquitectura SOA (Service Oriented Architecture), “establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios” [13]. La forma más habitual de implementar este tipo de arquitectura es mediante Servicios Web.

“Un Servicio WEB es una funcionalidad concreta que puede ser descubierta en la red, y que describe tanto lo que puede hacer como el modo de interactuar con ella; dichos servicios pueden realizar tareas muy concretas como realizar algún cálculo u obtener o extraer algún dato como por ejemplo el código de un cliente” [13].

Es importante mencionar que la característica principal de implementar una Arquitectura Orientada a Servicios, es permitir interconectar procesos entre las distintas aplicaciones que se manejan en una organización, pudiendo obtener interoperabilidad en cada uno de los procesos de negocio de ésta, y al mismo tiempo desarrollar dichos servicios de manera independiente a la tecnología en la que se encuentra implementada cada aplicación.

Uno de los enfoques que sirve como guía para la implementación de este tipo de arquitecturas, es el “Enfoque orientado a procesos de negocio”, considerado un diseño “TOP-DOWN” [14] [15], en donde cada funcionalidad del proceso de negocio explota sus recursos disponibles de manera flexible hacia otros.

En base a lo anteriormente expuesto, para lograr un diseño arquitectónico orientado a servicios dentro de SIRA, se realizó un análisis de los procesos clave o de negocio relacionados al registro, control y seguimiento de las participaciones en las

convocatorias ofrecidas por el CEL hacia los aspirantes; teniendo como objetivo principal la disagregación de los mismos en piezas modulares.

Posteriormente se diseñaron interfaces o contratos de cada uno de dichos servicios que se expusieron a través del tipo REST, obteniendo importantes características como:

- Bajo acoplamiento de componentes.
- Construcción de los servicios pensando en el cambio.
- Poder realizar ciclos de desarrollo incrementales por cada servicio.
- Capacidad de interoperar con otros sistemas o clientes consumidores independientemente de la tecnología que utilicen.

Es importante hacer notar que, de acuerdo a las características y el tamaño del proyecto, se adaptó el enfoque orientado a servicios como una arquitectura de diseño de solución que permitiera obtener las características anteriormente mencionadas, dentro de la funcionalidad del sistema SIRA, y que permitiera la posibilidad de interoperar con otros sistemas y procesos del Centro de Enseñanza LANIA; lo anterior servirá como base en miras de migrar hacia una arquitectura de servicios a nivel global, y que pudiera implementarse futuro en todos los sistemas del CEL.

3.2 Elección de una arquitectura SOA basada en servicios REST.

Cuando se habla de Arquitectura Orientada a Servicios, la tecnología comúnmente usada para desarrollar dichos servicios suele ser SOAP, puesto que ofrece una gama muy robusta de características que permiten establecer de manera muy precisa el intercambio de datos y el modo en el que interactuarán con los clientes o sistemas consumidores, sin embargo, no es el único modo de implementar este tipo de servicios. Una de las maneras que se utilizan demasiado hoy en día en sistemas basados en la *World Wide Web*, es el enfoque de Servicios de tipo REST (*Representational State Transfer*), cuyas características y enfoque puede ser

perfectamente aplicable dentro del mundo empresarial utilizando el protocolo HTTP [10].

Si bien no es objeto de este documento comparar ambos tipos de tecnología, las razones por las que se decidió utilizar un enfoque de tipo REST en el diseño de servicios dentro de SIRA fueron:

- Permitir definir de una manera muy ligera servicios de intercambio mediante peticiones HTTP, utilizando sus métodos definidos como son: POST, PUT, DELETE, PATCH y GET; y a la vez utilizar los códigos de respuesta nativos de HTTP (404,200,204,409, etc.).
- Basándose en la simplicidad para implementarlos que otras opciones como SOAP, puede traducirse en menor tiempo de desarrollo, siendo el tiempo un factor importante en este proyecto para la generación del prototipo funcional.
- Pensando en el tamaño del proyecto y por la finalidad de que éste sea consumido a través de la WEB, el enfoque REST proporciona características de comunicación *stateless* o sin estado, muy adaptables a este tipo de escenarios [10].
- El tipo de datos de intercambio se encuentra basado en formato JSON, el cual por sus siglas (*JavaScript Object Notation*), es interpretado de forma natural por JavaScript, lenguaje en el cual se basa el cliente *Frontend*, que se encarga de consumir dichos servicios REST, permitiendo realizar peticiones tipo AJAX, y obtener así la información de manera nativa para el cliente WEB.

3.3 Servicios identificados dentro del sistema SIRA.

En base a las características importantes que permite una arquitectura de tipo SOA, y de acuerdo a una etapa de análisis de requerimientos funcionales, no funcionales, así como de procesos clave dentro del contexto del sistema SIRA; se determinó disgregar los distintos procesos de negocio en los siguientes servicios tipo REST:

- 1) **Servicio REST de Usuarios:** Se encarga de realizar todos los procesos enfocados a la parte del registro, edición, eliminación de usuarios del sistema SIRA.
- 2) **Servicio REST de Catálogos:** De manera general, el servicio provee de todas las acciones necesarias para que una aplicación cliente pueda generar y consultar la base de catálogos del sistema, estableciendo así los objetos base para poder realizar los procesos operativos de SIRA.
- 3) **Servicio REST de Participaciones:** Dicho servicio se encarga del proceso núcleo o de negocio, que consiste en permitir generar la participación de un aspirante en una convocatoria, subir su documentación requisito, así como permitir consultar el estatus de dichas participaciones. Así también, provee de las funcionalidades necesarias para poder realizar los procesos de control y seguimiento que permitan conocer el estatus de dichas participaciones al personal por parte de LANIA, tanto de manera general por convocatoria como de manera específica por cada participación.
- 4) **Servicio REST de Notificaciones:** Este servicio se encarga de generar las notificaciones de correo que se realizan cuando ocurren eventos importantes dentro del funcionamiento de SIRA, tales como: El aviso de activación de cuenta para el nuevo aspirante, el aviso de registro del aspirante en alguna convocatoria, el aviso de cambio de estatus de la convocatoria del aspirante, entre otros.

3.4 Diagrama arquitectónico de SIRA.

En la página siguiente, se presenta un diagrama general del diseño propuesto de la arquitectura de SIRA, donde se propone una orientación a servicios en sus componentes. En dicho diagrama se puede apreciar la aplicación cliente WEB en la parte superior, la cual se conecta de manera independiente a través de los servicios ofrecidos por el backend, mediante el uso de peticiones tipo REST utilizando el protocolo HTTP, donde para cada petición se envía el token de autenticación (JWT, *Jason Web Token*) como parte de la cabecera de la petición.

Cada uno de los servicios se encarga de ciertos procesos bien definidos dentro del sistema, los cuales finalmente interactúan a través de sus capas hasta llegar al esquema de base de datos de SIRA, donde se encuentra persistida la información.

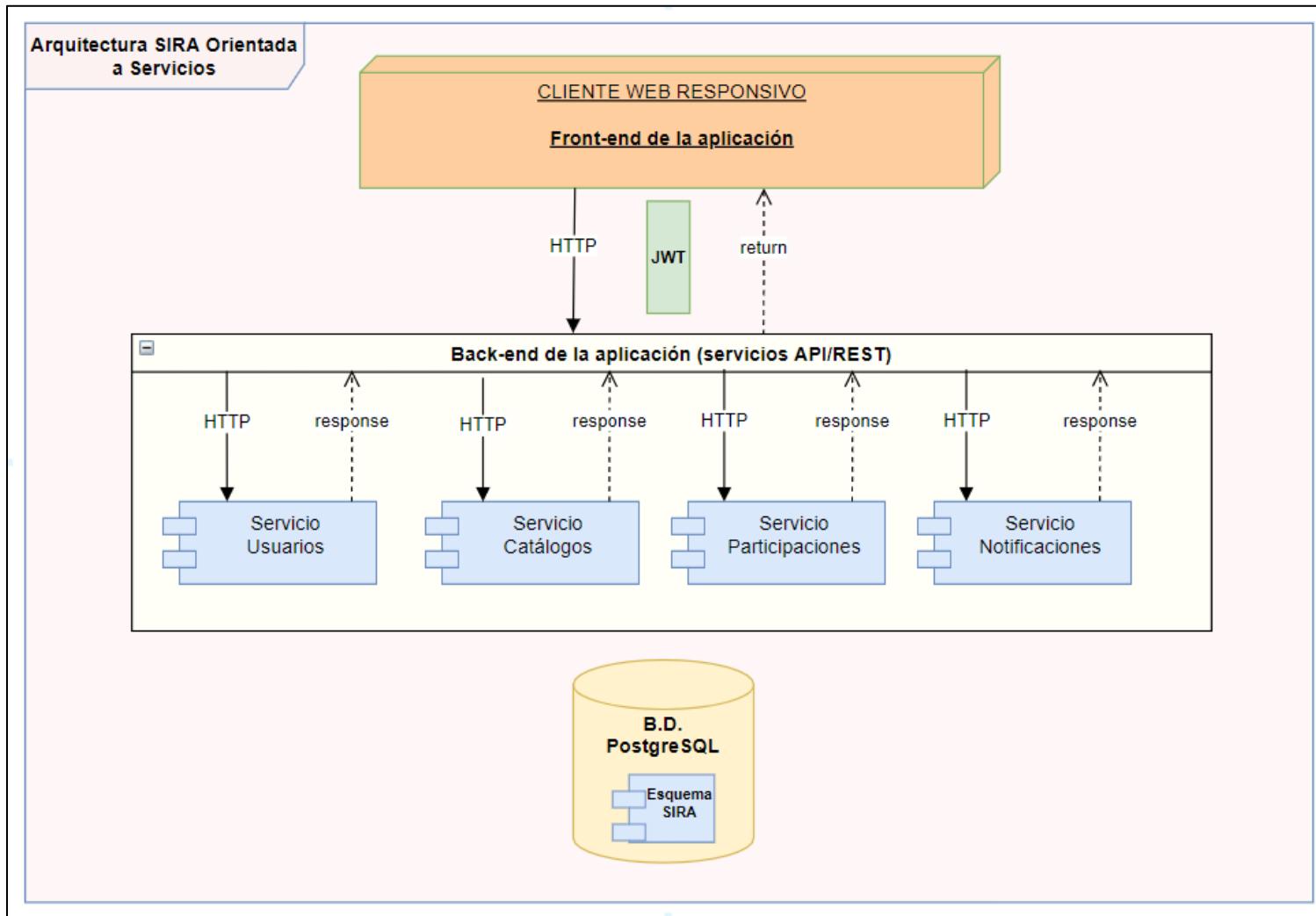


Figura 7. Diagrama Arquitectónico de SIRA.

3.5 Diseño general de los servicios en SIRA.

Como parte del diseño arquitectónico de los servicios de tipo REST implementados en SIRA, se tienen las siguientes características que aplican para todos los servicios diseñados:

- Son implementados bajo el enfoque tipo REST (*Representational State Transfer*), el cual utiliza como interfaz de comunicación peticiones basadas en el protocolo HTTP, mediante las que se generan las operaciones sobre los datos, cuyo formato de intercambio elegido es del tipo JSON (*Javascript Object Notation*) [10].
- La implementación de dichos servicios se realiza en un diseño arquitectónico de varias capas: la capa de acceso a datos y modelos, la capa de servicios de negocio, y la capa de Controladores REST. Cada una de estas capas cuentan con su interfaz correspondiente para establecer los contratos de comunicación entre ellas.
- Para todas las peticiones a dichos servicios, se genera un objeto de respuesta donde se establece el código HTTP response correspondiente, según el tipo de respuesta devuelta por la operación realizada del servicio solicitado.
- Se maneja un filtro de seguridad para controlar la autentificación de cada petición hacia todos los servicios. Dicho filtro está basado en el estándar JWT (*Jason Web Token*) con documento RFC 7519 [9], (el funcionamiento de dicho filtro se describe en la siguiente sección).
- Las URIS (*Unified Resource Identifier*), se encuentran diseñadas siguiendo las mejores prácticas de diseño para API's tipo REST [10]. De manera general se siguió el patrón:

/sira/{nombreservicio}/{recurso}/{identificador},

y para recursos relacionados el siguiente:

`/sira/{nombreservicio}/{recurso1}/{identificador}/{recurso2}`

- El intercambio de mensajes es mediante el formato de tipo JSON, sin embargo, puede configurarse de manera sencilla para aceptar formato XML o cualquier otro como PDF, CSV etc, simplemente configurando los tipos de respuesta del lado del servidor, y del lado del cliente realizar la petición con el tipo de archivo deseado en la cabecera de type/content file.
- Los códigos de respuesta HTTP manejados fueron los siguientes:
 - 200: OK. Operación exitosa.
 - 201: Objeto creado.
 - 204: Sin contenido.
 - 400: *Request* incorrecto.
 - 403: Prohibido.
 - 404: Objeto o *resource* no encontrado.
 - 500: Error de proceso del servidor.
- Métodos HTTP utilizados: POST, GET, PUT, PATCH, DELETE.

3.6 Capas de diseño de los servicios.

Dentro del diseño de cada servicio y tomando en consideración que uno de los objetivos del proyecto es poder desacoplar los componentes de negocio para dotar de una mayor facilidad en el mantenimiento de la aplicación, así como obtener reutilización de componentes; se decidió realizar un diseño de implementación dividido en capas, las cuales se describen en las siguientes secciones.

3.6.1 Capa de Modelado y Acceso a Datos.

En esta capa se realiza la implementación de todas las clases de tipo entidad, resultado del análisis de datos realizado en la fase 1 del proyecto. La idea de esta capa es contar con la representación de tipo objeto para cada uno de los modelos de datos, los cuales en su mayoría corresponden a un mapeo de cada tabla de la base de datos del esquema SIRA, los cuales son representados por clases entidad, y cada campo de tabla por atributos de la clase. (Para mayor referencia consultar el apartado de análisis de datos, “Diagrama de Clases”, del capítulo 1).

3.6.2 Capa de Servicio.

En esta capa se trata de separar todas las funcionalidades o procesos de negocio de cada servicio. Es aquí donde se encuentran implementadas las operaciones de manera detallada de acuerdo a las reglas de negocio de cada servicio. Dichas clases de servicio se encuentran implementadas a partir de sus correspondientes interfaces, las cuales definen la firma de los métodos para las operaciones a realizar y proporcionan el desacoplamiento necesario, ya que, de ser requerido, todas las implementaciones pudieran ser sustituidas en un futuro por otras, sin afectar a las demás capas de los servicios, siempre y cuando se respeten dichas interfaces.

Es a través de ésta capa que se tiene acceso a la capa de acceso a datos y se realizan las operaciones a las entidades a través de la interfaz de la capa DAO (*Data Acces Object*) o *Repository* correspondiente a cada entidad.

3.6.3 Capa de Controladores REST.

Esta es la capa de mas alto nivel, es decir, en ésta capa se definen los *paths* o URI's de acceso a las operaciones que estarán disponibles para la aplicación cliente o consumidora de dicho servicio. Aquí se definen todas las operaciones del servicio REST, las operaciones realizadas a través de las interfaces de la capa de servicio o negocio, y se conforma la respuesta de tipo HTTP, con el tipo de representación de

los objetos (JSON) como salida de datos de cada operación hacia la aplicación consumidora.

3.6.4 Diagrama de diseño arquitectónico de los servicios por capas.

En el siguiente diagrama se puede apreciar como se encuentran distribuidas las capas de diseño de cada servicio de SIRA:

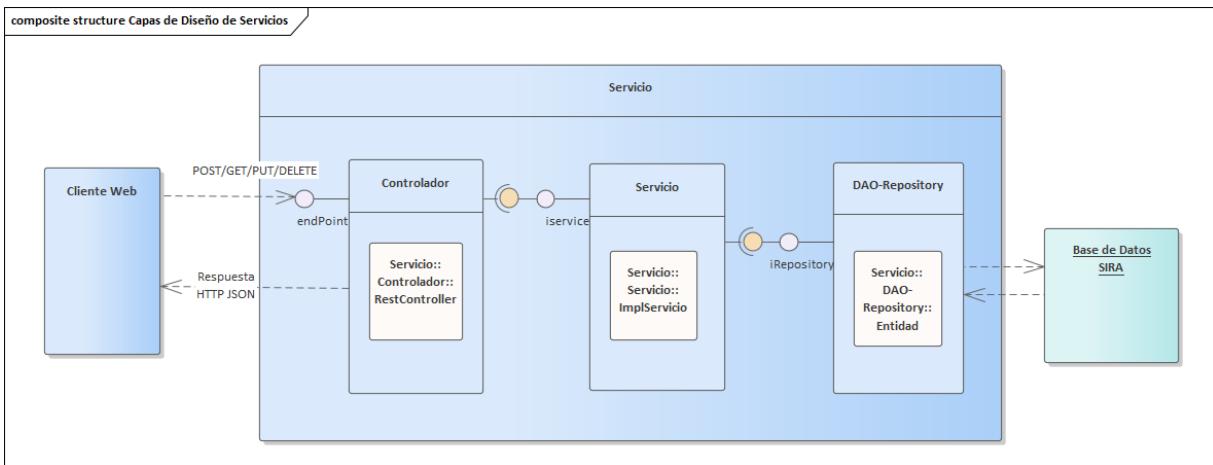


Figura 8. Diagrama de diseño arquitectónico por capas.

3.7 Filtro de seguridad en los servicios.

Una de las cuestiones obligatorias a controlar al realizar un diseño con una Arquitectura Orientada a Servicios, es la protección y control de la información que exponen, ya que al hacer públicos sus *endpoints* para ser consumidos por algún cliente, se corre un gran riesgo de poder ser vulnerados si no se dota de los protocolos de seguridad adecuados.

Bajo este precepto, se tiene que “*la seguridad no solo se determina mediante el usuario y contraseña, sino que además se debe verificar qué funciones y permisos*

están establecidos en cada usuario del sistema para asegurar que el acceso a los recursos de cada servicio es legítimo” [11].

Uno de los problemas que se tienen al momento de implementar este tipo de arquitecturas, es que las peticiones que se realizan a dichos servicios son de tipo asíncronas y *stateless* (sin estado). Al ser de esta naturaleza, el servicio REST no puede recordar las credenciales del usuario cada que se realiza una nueva petición, puesto que no existe una sesión activa HTTP como lo es en arquitecturas tradicionales tipo MVC o Cliente-Servidor.

Tomando en cuenta dicha situación, además del típico control de credenciales basado en usuario y contraseña, se tomó la decisión de implementar un filtro en cada una de las peticiones HTTP que llegan a los servicios de SIRA.

Dentro de las opciones que se contemplaron para aplicar seguridad al momento de autenticar y autorizar las peticiones de los clientes a los servicios fueron:

- Autentificación básica.
- Autentificación basada en *token*.
- Autentificación basada en clave API.
- OAuth 2.0 (Autorización abierta).

Después de analizar dichas opciones, se tomó la decisión de implementar el filtro con Autentificación basada en *token*, ya que a diferencia de la primera (Autentificación básica), es menos vulnerable a un ataque de *tipo man-in-the-middle* y, comparado con la de basada en clave API y OAuth 2, es un poco mas simple de implementar [11], por lo que siendo estrictos a los objetivos del proyecto, se prefirió invertir ese tiempo de desarrollo en implementar las funcionalidades que están relacionadas al proceso de negocio de la aplicación, llegando a la conclusión de que se ofrece un buen nivel de seguridad versus complejidad de implementación. No obstante, es de resaltar que, por la misma naturaleza de la arquitectura de los servicios, pudieran implementarse otros protocolos en el futuro sin ningún problema.

3.7.1 Autenticación basada en Jason Web Token.

Como uno de los estándares más conocidos para la autenticación basada en *tokens*, se tiene JWT (*Jason Web Token*), el cual se encuentra basado en el RFC 7519 [9].

Los *tokens* JWT son un tipo de *token* que engloba cierta estructura, la cual puede ser desencriptada por el servidor y de esa manera permitir autenticar al usuario en la aplicación.

Un *token* JWT es en sí una cadena dividida en 3 secciones separadas por un punto. Estas tres secciones tienen un significado:

- HEADER: La primera parte, corresponde a los *Header* o cabeceras, y se almacena por lo general el tipo de *token* y el algoritmo de encriptación.
- PAYLOAD: La segunda parte, contiene los datos que identifican al usuario, como puede ser su ID, nombre de usuario, etc.
- FIRMA: La tercera parte es la firma digital, la cual se genera con las secciones anteriores y sirve para validar que el contenido no haya sido alterado.

Dentro de SIRA el *token* JWT se encuentra conformado por las partes anteriormente mencionadas. El siguiente es un ejemplo de *token* JWT de SIRA:

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJtYWljNzAZ21haWwuY29tliwiYXV0aG9yaXRpZXMiOlt7ImF1dGhvcmI0eSI6IIVTRVIifV0sImIhdCI6MTYyOTUwNjI4NywiZXhwIjoxNjMwNjQ1MjAwfQ.jKGfuPkq5_AFp6-3b5QEnu9WL82ZVqsj7Xx0Cdahd4yr2czja9qiXtoyr2mGMM.

El cual, al decodificarse, cada parte se interpretaría como sigue:

- **Header:** Establece el tipo de algoritmo de cifrado, en este caso SHA-384 [12].

{ "alg": "HS384" }

- **Payload:** Establece los datos del usuario, permisos y expiración del token.

```
{
  "sub": "mlanda@lania.edu.mx",
  "authorities": [
    {
      "authority": "ASPIRANTE"
    }
  ],
  "iat": 1629506287,
  "exp": 1630645200
}
```

- **Firma:** Establece la firma digital de verificación.

```
HMACSHA384(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) secret base64 encoded
```

3.7.2 Diagrama del proceso de Autenticación de usuarios basado en JWT.

De manera general, para garantizar la autenticación de usuarios se implementó el siguiente proceso:

- 1 El usuario se registra y obtiene sus credenciales (usuario y contraseña)
- 2 El usuario se autentifica, el cliente recibe el *token* JWT y lo guarda para las siguientes peticiones REST.
- 3 El cliente usa el *token* para las siguientes peticiones REST.

En el siguiente diagrama se aprecia el flujo del proceso de autenticación de usuarios implementado:

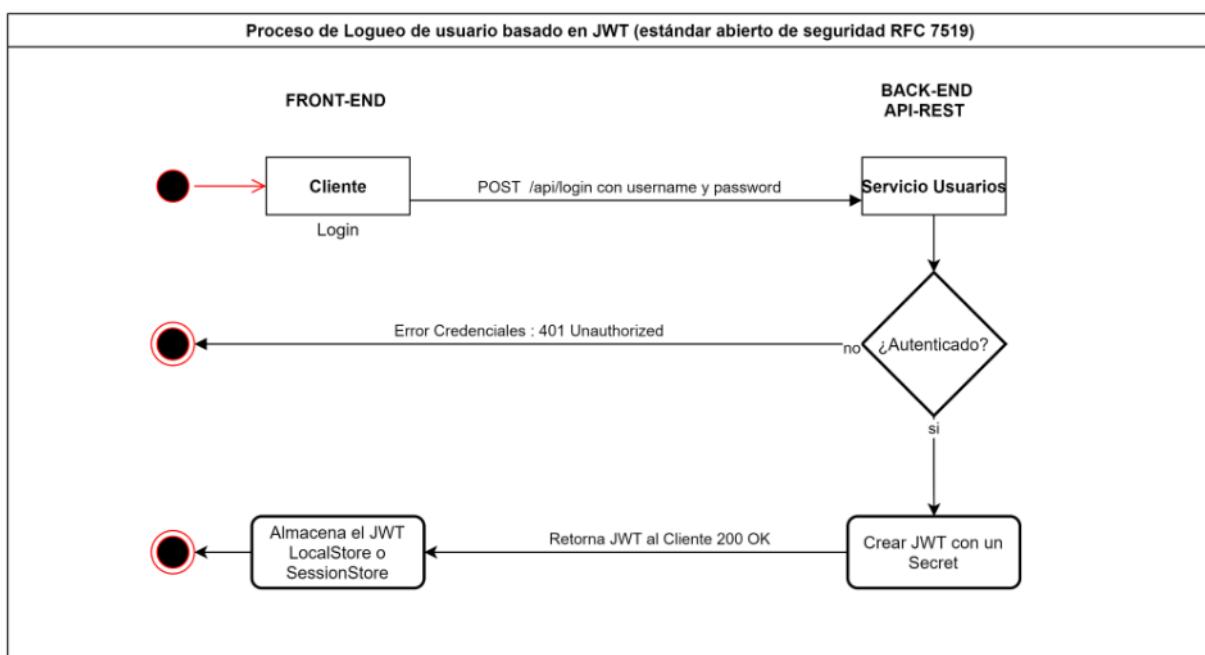


Figura 9. Proceso de autenticación de usuario basado en JWT.

3.7.3 Diagrama del proceso de acceso a un recurso REST basado en JWT.

Una vez obtenido el *token* por el cliente, se utiliza el mismo para generar cualquier tipo de petición hacia los servicios REST de SIRA, enviándose dentro del campo “*Authorization*” perteneciente al encabezado de la petición HTTP.

En la Figura 10, se describe de manera detallada el proceso mediante el siguiente diagrama:

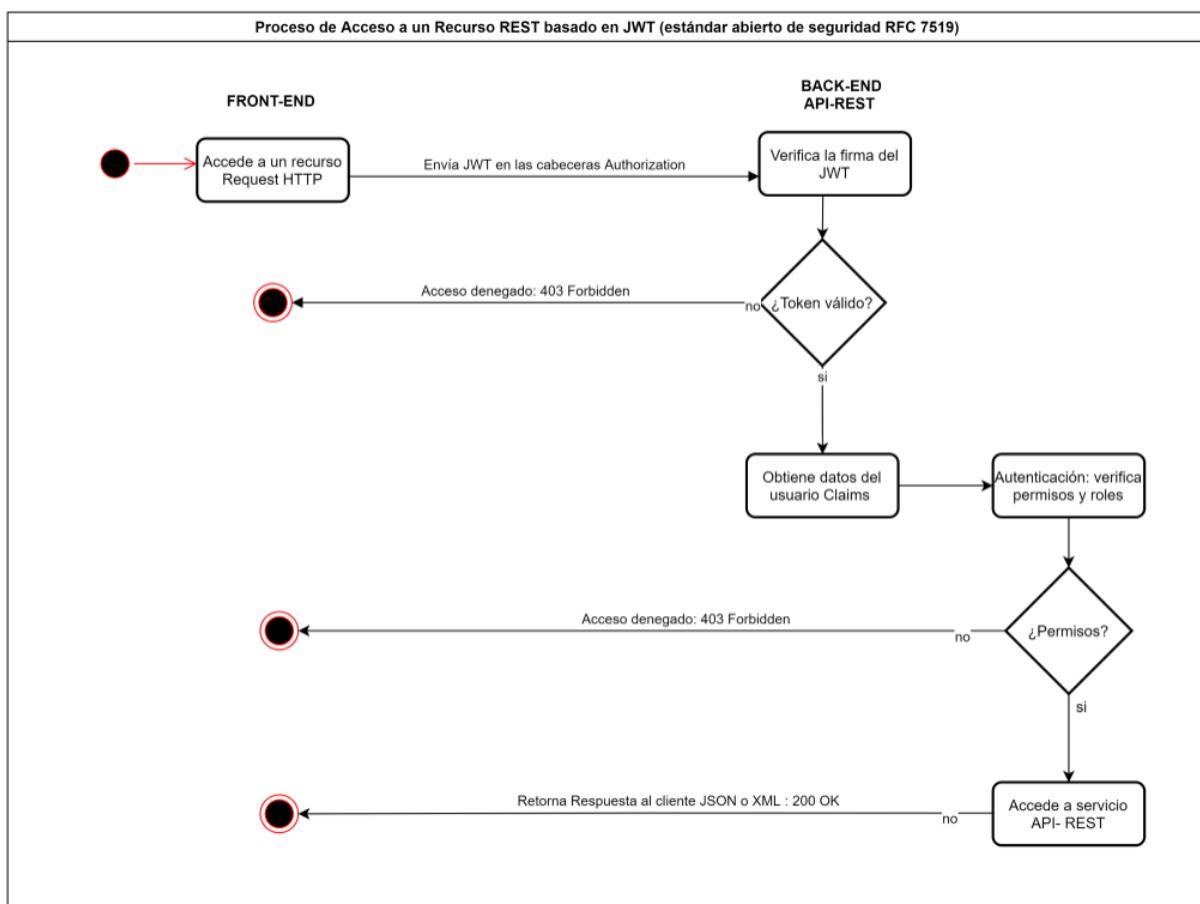


Figura 10. Proceso de acceso a un recurso REST basado en JWT.

3.8 Servicio REST de usuarios.

En las subsecciones siguientes se describen las funcionalidades del servicio versus casos de uso, las acciones y métodos REST, así como los diagramas de clases que lo conforman y sus relaciones.

3.8.1 Alcance de funcionalidades del servicio contra casos de uso.

Como se mencionó anteriormente, el Servicio de Usuarios, tiene como principal responsabilidad el permitir el registro, modificar y eliminar usuarios, así como el proceso de confirmación y habilitación de cuentas de usuario registradas. Dentro de este servicio se abarcan los casos de uso:

- “CU-01 Crear cuenta de usuario aspirante”: El aspirante ingresa sus datos generales y recibe correo con su cuenta de usuario para poder confirmar su cuenta. Una vez confirmada podrá ingresar al sistema SIRA.
- “CU-02 Ingresar al Sistema”: El usuario ingresa al sistema mediante sus credenciales de usuario y contraseña.
- “CU-13 Consumir datos generales de aspirantes”: SICEL consume datos generales de los aspirantes.

Para ver la especificación de dichos casos de uso consultar el ANEXO A.

3.8.2 Diseño de acciones y métodos del Servicio REST Usuarios.

Para generar el diseño de las acciones y métodos que tiene disponibles el Servicio REST de Usuarios, se tomó en cuenta el análisis de casos de uso del proceso de registro de Usuario, como el análisis de datos descrito en el capítulo 2, donde se tomaron en cuenta las siguientes entidades o modelos: Usuario, Aspirante y Empleado, las cuales se describen en el siguiente diagrama:

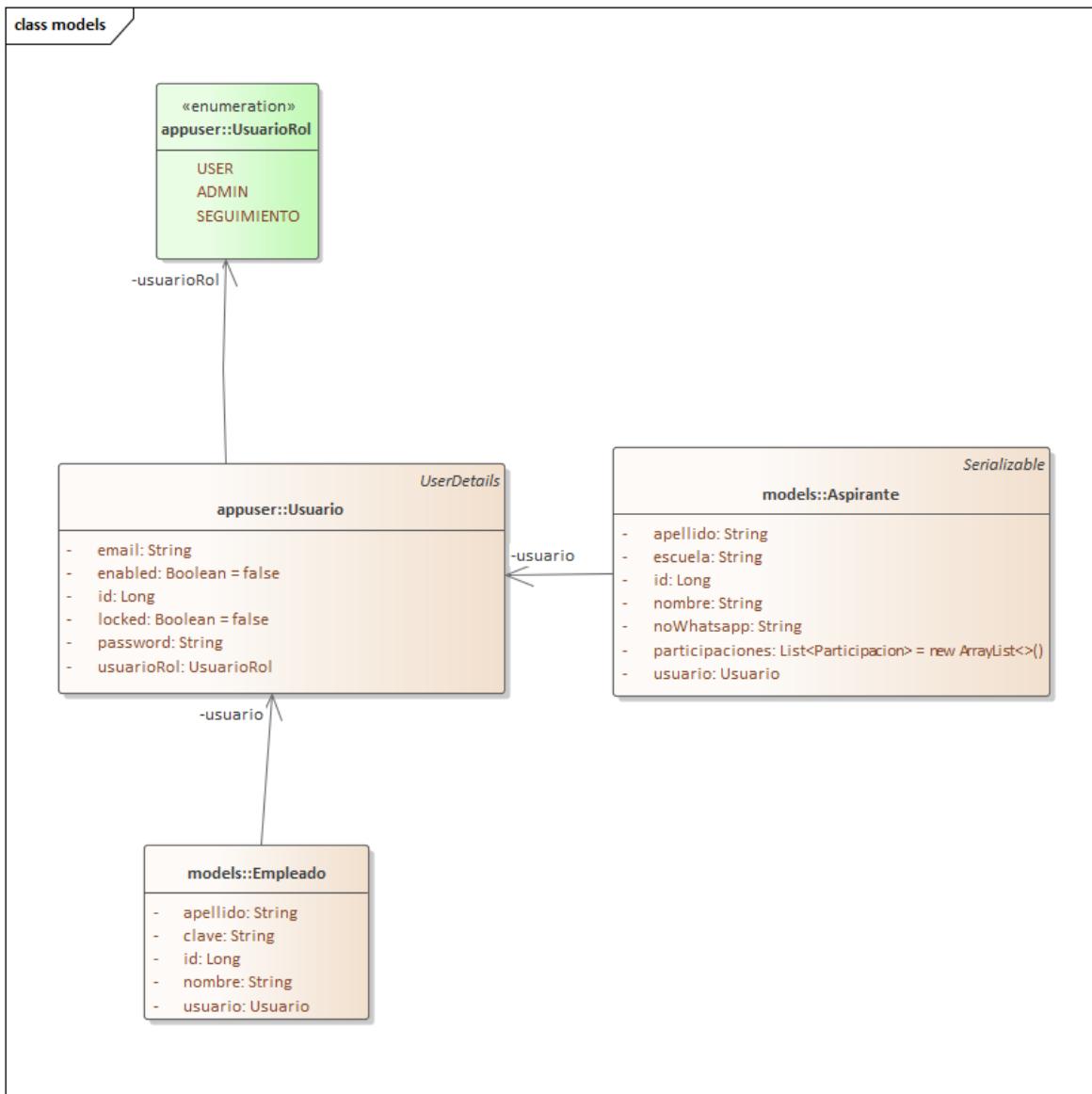


Figura 11. Modelos del Servicio Usuario.

También se realizó el diseño de modelos y clases auxiliares al proceso de registro de usuarios, como son: el controlador de registro, la capa de servicio de registro, el *request* de tipo de registro, las cuales se describen a continuación:

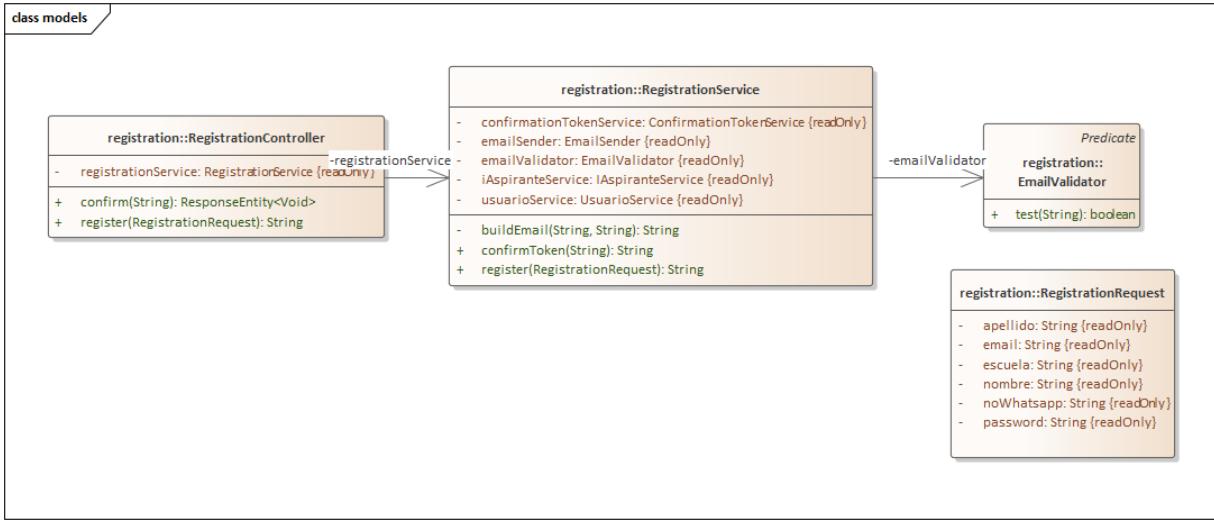


Figura 12. Modelos y clases auxiliares del proceso de registro de usuarios.

Se diseñó el controlador REST “*RegistrationController*”, el cual expone los *endpoints* descritos en la siguiente sección.

Se diseñó una capa de servicio con procesos propios para el registro y en el cual se engloba el uso de interfaces con la capa de acceso a datos (DAO) o repositorios Aspirante y Empleado.

Se diseñó el uso de Repositorios de JPA (*Java Persistent API*) para realizar la inserción, consulta y borrado de datos de la base de datos del esquema SIRA.

3.8.3 Descripción de *endpoints* del servicio de usuarios.

En la siguiente tabla se encuentran descritos los *endpoints* de tipo REST disponibles en el servicio de usuarios:

Método HTTP	Endpoint Prefijo: /sira/usuarios	Entidades	Acción	Entrada	Respuesta
POST	/sira/usuarios/aspirantes	Usuario Aspirante	Alta de usuario Aspirante	Request Registro	token para confirmar registro
GET	/aspirantes	Usuario Aspirante	Obtener los usuarios aspirantes	N/A	Lista de usuarios aspirantes
PUT	/aspirantes/{id}	Usuario Aspirante	Modificar usuarios aspirantes	id Aspirante	Usuario Modificado
DELETE	/aspirantes	Usuario Aspirante	Eliminar usuario aspirante	id Aspirante	mensaje exitoso
POST	/aspirantes/confirmar	Usuario Aspirante	Confirmar cuenta de usuario aspirante	token de registro	liga de cuenta activada.
POST	/empleados	Usuario Empleado	Alta de usuario Empleado	Request Registro	Usuario Empleado
GET	/empleados	Usuario Empleado	Obtener los usuarios empleados	N/A	Lista de usuarios empleados
PUT	/empleados/{id}	Usuario Empleado	Modificar usuarios empleados	id Empleado	usuario modificado
DELETE	/empleados	Usuario Empleado	Eliminar usuario empleado	id Empleado	mensaje exitoso

Tabla 1. *Endpoints* del servicio REST usuarios.

3.9 Servicio REST de catálogos.

En este apartado se describe el servicio de catálogos, el cual se encarga de controlar todos los catálogos de programas educativos, convocatorias, requisitos y sus relaciones. Se describe también sus acciones y métodos, así como los diagramas de clases correspondientes. Este servicio permite todas las operaciones de tipo CRUD (*Create, Read, Update, Delete*), de cada entidad anteriormente mencionada y poder contar con toda la estructura base para poder controlar su persistencia.

3.9.1 Alcance de funcionalidades del servicio contra casos de uso.

Dentro de este servicio se contemplan los casos de uso:

- “CU-03 CRUD de Programas Educativos”.
- “CU-04 CRUD de Convocatorias”.
- “CU-14 CRUD de Requisitos”.
- ”CU-15 Asignar requisitos a convocatorias”.
- “CU-05 Consultar convocatorias disponibles”.

Para ver la especificación de dichos casos de uso consultar el ANEXO A.

Dicho servicio se encarga de las funcionalidades de alta, baja y modificación de programas educativos, convocatorias y requisitos, así como de permitir relacionar las convocatorias con los requisitos deseados. Permite establecer la base de objetos-entidad para poder realizar las operaciones de negocio del sistema.

3.9.2 Diseño de acciones y métodos del Servicio REST Catálogos.

De igual manera que en el servicio de usuarios, para el diseño de este servicio se tomó en cuenta el análisis de casos de uso, procesos clave y datos del capítulo 2, específicamente en el diseño de clases para este servicio se tienen las siguientes:

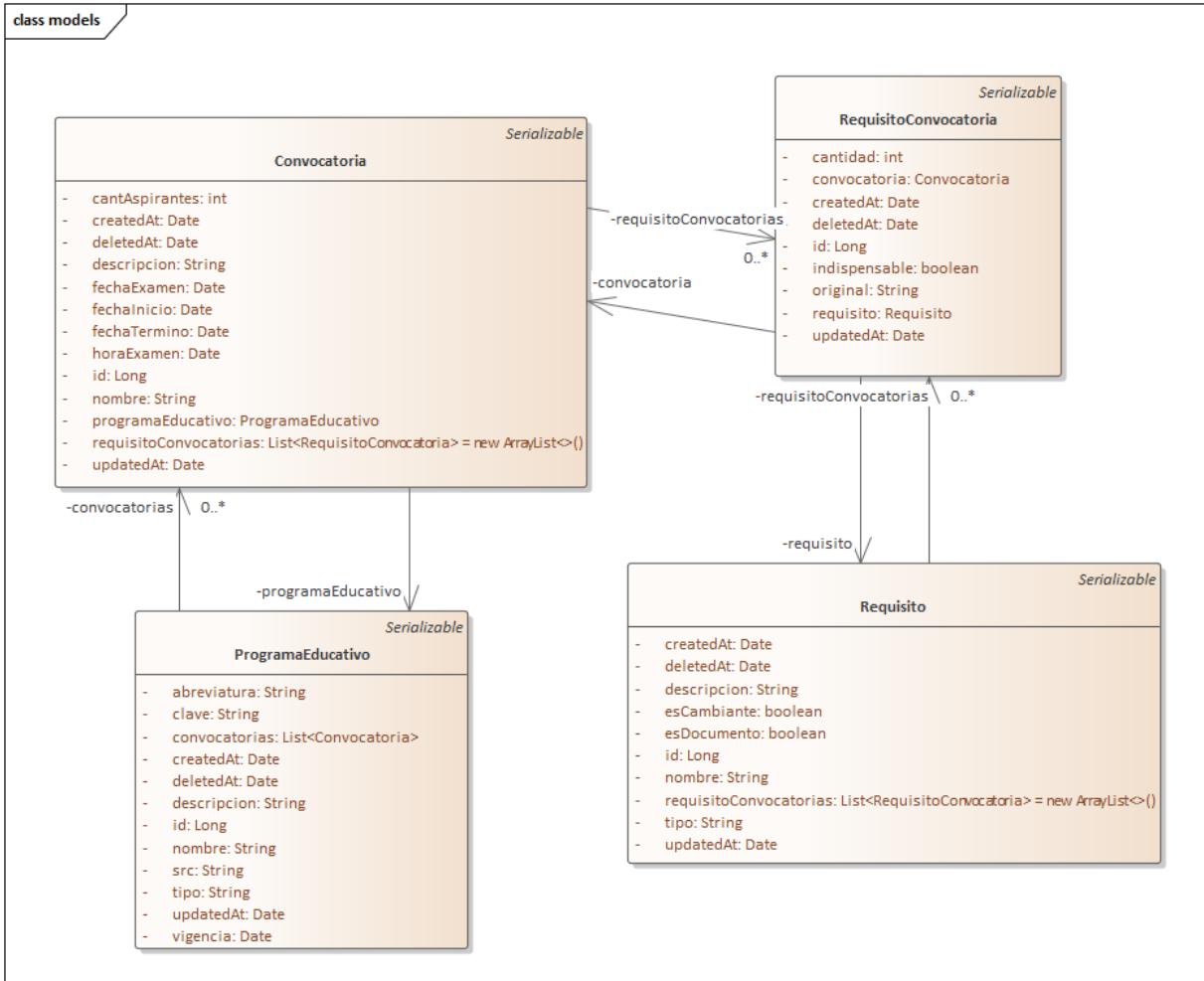


Figura 13. Diseño de clases del servicio de catálogos.

Al igual que en el servicio anterior, se definieron clases auxiliares para las capas de diseño como son el controlador, la capa de servicio y los DAO's o repositorios que acceden al esquema de base de datos de SIRA.

Es de remarcar que dentro del análisis de datos se determinó que existe una relación de tipo muchos a muchos entre la entidad “*Requisito*” y la entidad “*Convocatoria*”, por lo que se definió una clase entidad auxiliar como intermedia, para poder mapear adecuadamente la relación entre ellas.

3.9.3 Descripción de endpoints del Servicio de Catálogos.

En la siguiente tabla se describen las operaciones y los endpoints de tipo REST que ofrece el servicio de catálogos.

Método HTTP	Endpoint Prefijo: /sira/catalogos	Entidades	Acción	Entrada	Respuesta
POST	/peducativos	Programa Educativo	Alta de P.E.	Progr ama Educa tivo	Programa Educativo
GET	/peducativos	Programa Educativo	Obtener los P.E.	N/A	Lista de P.E.
PUT	/peducativos/{id}	Programa Educativo	Modificar P.E.	id P.E.	Programa Educativo mod.
DELETE	/peducativos/{id}	Programa Educativo	Eliminar P.E.	id P.E.	mensaje exitoso
POST	/convocatorias/peducativos/{id}	Convocatoria Prog. Educativo	Alta de Conv.	Convo catori a e id P.E.	Convocatoria
GET	/convocatorias	Convocatoria	Obtener Convs.	N/A	Lista de convocatorias
PUT	/convocatorias/{id}/peducativos/{id}	Convocatoria Prog. Educativo	Modificar Conv.	id convoco catoria	conv. modificada
DELETE	/convocatoria/{id}	Convocatoria	Eliminar Conv.	id convoco catoria	mensaje exitoso
POST	/requisitos	Requisito	Alta de Requisito	Requisito	Requisito
GET	/requisitos	Requisito	Obtener req.	N/A	Lista de Req.
PUT	/requisitos/{id}	Requisito	Modificar req.	id Requisito	Requisito mod.
DELETE	/requisitos/{id}	Requisito	Eliminar req.	id Requisito	mensaje exitoso
POST	/convocatorias/{id}/requisitos/{id}	Convocatoria Requisito	Agregar Requisito a Conv.	id convoco catoria	lista de requisitos

				id requisi to	
DELETE	/convocatorias/{id}/requisi tos/{id}	Convocat oria Requisito	Eliminar req. de conv.	id convo catori a id requisi to	mensaje exitoso
GET	/convocatorias/{id}/requisi tos	Convocat oria Requisito	Obtener req. de conv.	id convo catori a	lista de requisitos de convocato ria

Tabla 3. *Endpoints del servicio de catálogos.*

3.10 Servicio REST de participaciones

En las siguientes subsecciones, se describe las funcionalidades del servicio de participaciones, así como las clases que lo conforman. Al igual que los demás servicios se describe cada una de las operaciones tipo REST disponibles.

3.10.1 Alcance de funcionalidades contra casos de uso.

Se trata del servicio *core* que realiza las funciones más sustanciosas relacionadas a la lógica de negocio, y que a su vez tiene que ver con todos los componentes de los demás servicios. Por un lado, interactúa con los componentes del servicio de usuarios al consultar el aspirante que desea participar en una convocatoria, y, por otro lado, con el Servicio de Catálogos al relacionar los requisitos de la convocatoria con la participación del aspirante. Finalmente, también interactúa con el servicio de notificaciones puesto que se realizan diversas notificaciones vía correo, derivadas de algunos eventos que ocurren en el proceso de la participación del aspirante.

Para lograr identificar los procesos clave de este servicio se consideraron los casos de uso siguientes:

- CU-06 “Registrar participación en convocatoria”: El aspirante registra su participación en alguna convocatoria disponible.
- CU-07 “Subir documentación en convocatoria”: El aspirante registra su documentación en la convocatoria donde participa.
- CU-08 “Verificar documentación del aspirante”: El administrador verifica la documentación del aspirante en la convocatoria.
- CU-09 “Consultar el estado de las convocatorias del aspirante”: El aspirante consulta el estatus de su participación en las convocatorias.
- CU-10 “Visualizar el estado general de las convocatorias”: El usuario seguimiento o administrador consulta el estatus de las convocatorias.

3.10.2 Diseño de acciones y métodos del Servicio REST de Participaciones.

Para el diseño de clases entidad, se tomó como base el análisis de datos realizado en el capítulo 2, en donde se identificaron las clases entidad de “Participación” y “ParticipacionRequisitoConvocatoria”. Cabe señalar que la clase “Participación” logra abstraer la relación muchos a muchos entre la clase “Aspirante” y la clase “Convocatoria”, y ésta a su vez se relaciona muchos a muchos con la entidad “ParticipacionRequisitoConvocatoria”, la cual encapsula la relación muchos a muchos con cada “RequisitoConvocatoria”; el propósito de esta última clase es poder realizar la persistencia de cada requisito de la convocatoria con los requisitos entregados por parte del aspirante, es decir el estado de la participación.

A continuación, se muestra un diagrama con las clases anteriormente mencionadas y su relación:

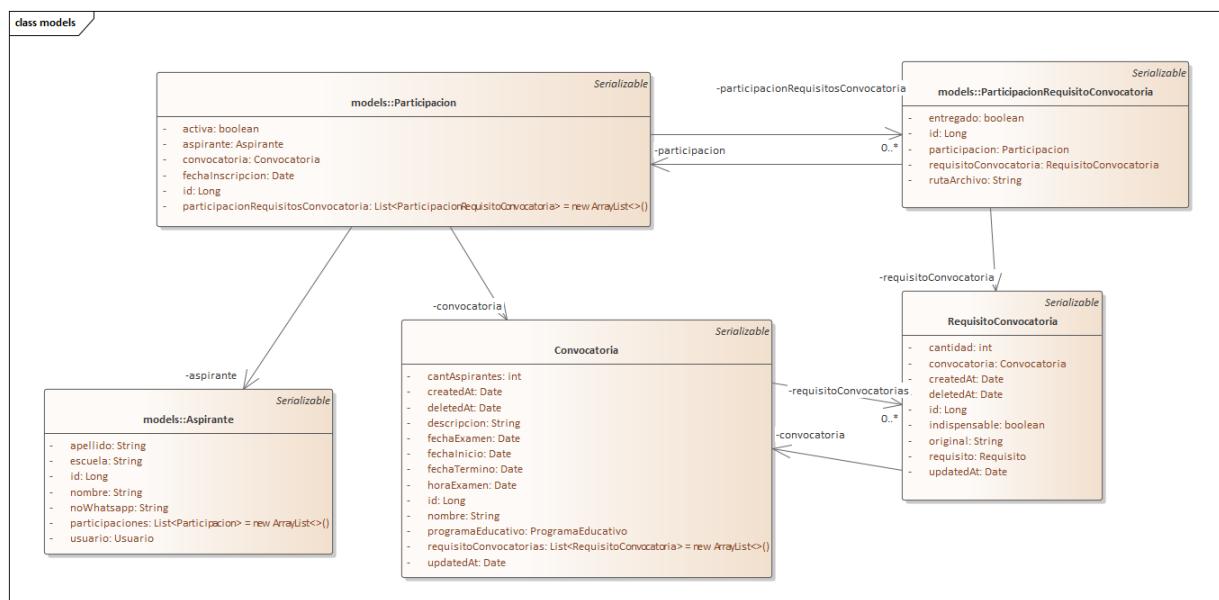


Figura 14. Diagrama de clases de Participaciones y sus relaciones.

3.10.3 Descripción de *endpoints* del Servicio de Participaciones.

En la siguiente tabla se describen las operaciones y *endpoints* tipo REST disponibles en el servicio de participaciones:

Método HTTP	Endpoint Prefijo: <code>/sira/participaciones</code>	Entidades relacionadas	Acción	Entrada	Respuesta
GET	<code>/aspirantes/{id}</code>	Aspirante Participacio n	Obtener las Participaci ones del Aspirante	id Aspirant e	Lista de Participaci ones del aspirante
POST	<code>/aspirantes/{id}/convoca torias/{id}</code>	Aspirante Convocatori a Participacio n Participacio n-Requisito- Convocatori a	Registrar la participaci on, crea los Participaci on- Requisito- Convocat oria	id Aspirant e id Convoca toria	Mensaje exitoso
PUT	<code>/aspirantes/{id}/participa cionrcn/{id}</code>	Aspirante Participacio n-Requisito- Convocatori a	Actualiza el Requisito Convocat oria de la Participaci ón (subir documento)	id Aspirant e id PartReq Conv archivo pdf	Mensaje estatus completad o
DELETE	<code>/aspirantes/{id}/participa cion/{id}</code>	Aspirante Convocatori a Participacio n Participacio n-Requisito- Convocatori a	Eliminar una Participaci ón del Aspirante	id Aspirant e id participa cion	mensaje exitoso

GET	<code>/{id}/aspirantes/{id}</code>	Aspirante Participación Participación-Requisito-Convocatoria	Obtener una participación del aspirante y su detalle	id Aspirante id Participación	La participación del aspirante: (lista de requisitos de la participación con estatus).
GET	<code>/aspirantes/</code>	Participación	Obtiene las participaciones de todos los aspirantes	N/A	Lista de Participaciones de todos los aspirantes
GET	<code>/aspirantes/convocatorias/{id}</code>	Aspirante Convocatoria Participación	Obtiene las participaciones de todos los aspirantes de una convocatoria	id Convocatoria	Lista de Participaciones de todos los aspirantes de una convocatoria
GET	<code>/aspirantes/convocatorias</code>	Aspirante Convocatoria Participación	Obtiene las participaciones de todos los aspirantes de todas las convocatorias	N/A	Lista de Participaciones de todos los aspirantes de todas las convocatorias

Tabla 4. Descripción de endpoints del Servicio de Participaciones.

3.11 Servicio REST de notificaciones.

El Servicio de Notificaciones encapsula las notificaciones vía correo electrónico que se envían al aspirante en relación a diversos eventos que ocurren en el sistema, tales como:

- La liga de confirmación de su cuenta de usuario: Ocurre cuando el usuario aspirante se registra en la plataforma, en dicho correo se le envía una liga con un *token* generado por el sistema, para confirmar su cuenta.
- Aviso de la participación al registrarse en una convocatoria: En este correo se le envía un aviso de que se encuentra registrado en una convocatoria, y se le envían los detalles de la misma.
- Aviso de que el estatus de su participación se encuentra completada: En este correo se le notifica al usuario que sus documentos han sido revisados y se encuentran en estatus completado.

3.11.1 Alcance de funcionalidades contra casos de uso.

Dentro del servicio de notificaciones se contemplan los siguientes casos de uso:

- CU-01 Crear cuenta de usuario aspirante.
- CU-06 Registrar participación en convocatoria.
- CU-08 Verificar documentación del aspirante

En dichos casos de uso, existe un momento en donde se envían notificaciones al usuario aspirante vía correo electrónico. El primero dentro del CU-01, al momento de registrarse como aspirante dentro de SIRA, donde se envía un correo de confirmación el cual se realiza a través del servicio de notificaciones. El segundo dentro del CU-06, al momento que el aspirante se registra en la convocatoria deseada, donde se envía una notificación con los detalles de la participación a través del servicio de notificaciones. Y finalmente, en el CU-08 al momento de que los documentos requisito de la participación del aspirante se encuentran validados, el sistema envía un correo electrónico para dar a conocer el estatus de la participación al aspirante.

3.11.2 Diseño de acciones y métodos del servicio REST de notificaciones.

Dentro del diseño del servicio de notificaciones, se implementó la clase EmailService y la interfaz EmailSender, las cuales se encargan de construir el correo que se enviará al usuario, así como de contar con el método send para el envío y los métodos para cada tipo de evento mencionados en el apartado anterior. Los diagramas de clase realizados en el proceso de análisis, se despliegan en la siguiente figura:

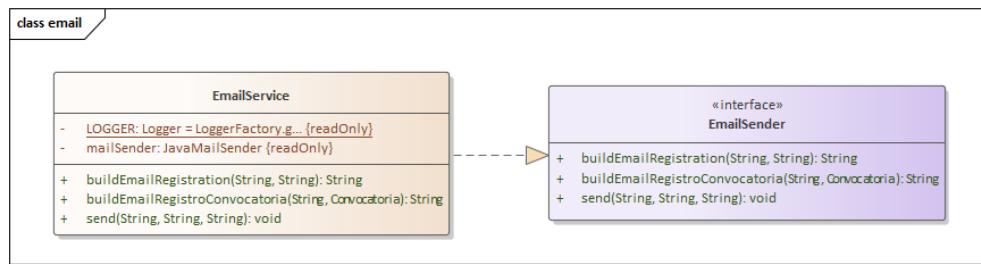


Figura 15. Diagrama de clase del servicio notificaciones para envío de correos.

Finalmente, se generó la clase de tipo controller NotificacionesRestController, donde se define el método enviarCorreoParticipación, el cual hace uso de los servicios de emailSender y usuarioService del servicio de usuarios, para realizar el envío de correos a los usuarios de SIRA.

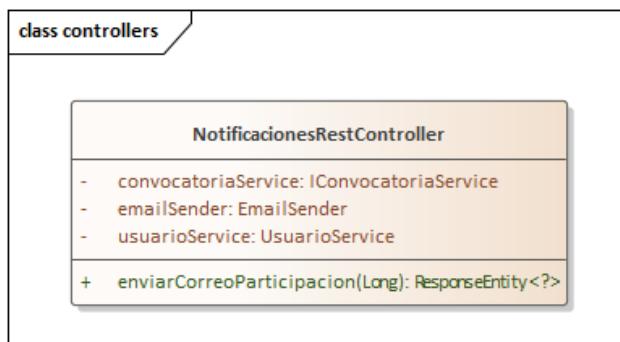


Figura 16. Diagrama de clase del controlador de notificaciones.

3.11.3 Descripción de *endpoints* del Servicio de Notificaciones.

En la tabla siguiente se describen las operaciones tipo REST disponibles en el servicio de notificaciones:

Método HTTP	Endpoint Prefijo: <code>/sira/notificaciones</code>	Entidades relacionadas	Acción	Entrada	Respuesta
POST	<code>/convocatorias/{idConvocatoria}</code>	Convocatoria Usuario	Enviar correo de la participación a la convocatoria	<code>idConvocatoria</code>	Mensaje exitoso
POST	<code>/aspirantes/{id}</code>	Aspirante Usuario	Enviar correo de confirmar usuario	<code>id Aspirante</code>	Mensaje exitoso
POST	<code>/aspirantes/{id}/participacion/{id}</code>	Aspirante Participacion	Enviar correo de estatus de participación	<code>idAspirante, idParticipacion</code>	Mensaje exitoso

4. Análisis y diseño de la interfaz de usuario responsiva.

4.1 Elección de la tecnología para la interfaz de usuario.

Dentro de los requerimientos no funcionales del sistema SIRA, se planteó la necesidad de realizar una interfaz de usuario que pudiera permitir una adecuada visualización al momento de consultarse desde un dispositivo móvil como *smartphones* o tabletas.

Tecnologías como HTML 5, Hojas de Estilo en Cascada (CSS) y el lenguaje JavaScript, son la base para poder implementar la estructura, el diseño y el comportamiento de una aplicación que es consumida por medio de un navegador WEB; sin embargo, el poder realizar un diseño responsive y adaptable para dispositivos móviles implementando solo estas tres tecnologías requiere de un gran esfuerzo de desarrollo.

Para mermar la problemática arriba descrita, se decidió optar por implementar una tecnología tipo Framework en la parte del *Front-end* de la aplicación, la cual proveyera de ciertas características, herramientas y componentes que permitieran hacer más rápido y sencilla la implementación básica de la interfaz de usuario de SIRA.

Alineado a dicho requerimiento, se realizó un análisis sobre las opciones que existen para el desarrollo rápido de interfaces responsivas hoy en día, en el que por un lado se tenía la opción de utilizar ciertas librerías que ayudan en el diseño de interfaces web: opciones como *Bootstrap* o *MaterialDesign*, que se basan en buenas prácticas y estándares de diseño de páginas web; sin embargo, realizando un análisis de opciones más robustas, se determinó utilizar el *Framework VuetifyJs*, el cual, en el aspecto de diseño se encuentra basado en *Material Design* (tecnología desarrollada por Google), y provee además de componentes orientados al desarrollo de interfaces reactivas, tal como lo hace *Angular* o *ReactJs*, dotando así de mejores características a la interfaz de usuario.

Es de remarcar que *VuetifyJs*, además de permitir estructurar y realizar un diseño responsive mediante clases o reglas de manera muy sencilla, utilizando sistemas de rejilla como lo hace Bootstrap; también provee de comportamientos reactivos en el dinamismo y actualización de componentes de la vista de la aplicación, ya que maneja un DOM (*Document Object Model*) virtual, tal como lo hace *ReactJs*.

Adicional a lo anterior, se trata de un *framework* progresivo, es decir no es necesario utilizar todas las herramientas que provee desde un inicio, sino que se van utilizando conforme la necesidad de complejidad de la aplicación lo requiera; lo que se traduce también en una curva de aprendizaje más pequeña y por ende en menos tiempo de desarrollo [16].

4.2 Diseño y estructura de los componentes de la interfaz.

El diseño de los componentes de la interfaz se realizó alineándose a los servicios identificados en el capítulo anterior; por lo que se tomaron en cuenta cada uno de los Casos de Uso descritos en el anexo A del presente documento.

Como primera etapa se realizaron algunas propuestas de diseño a manera de *Wireframes* utilizando la herramienta libre “*NinjaMock*”, esto para tener una idea general del diseño, elementos visuales y elementos necesarios en cada una de las vistas de la aplicación, dividiéndolas principalmente en las siguientes:

- Layout General de la aplicación.
- Componente de vista de Acceso y Registro de Aspirante.
- Componente de CRUD de Programas Educativos
- Componente de CRUD de Convocatorias.
- Componente de CRUD de Requisitos.
- Componente de Participaciones del Aspirante.
- Componente de Dashboard o tablero de estadísticas.

4.2.1 Wireframes y diseño.

A continuación, se muestran algunos de los *wireframes* realizados durante el proceso de diseño del cliente WEB:

En la siguiente figura se aprecia la pantalla propuesta mediante el *wireframe* de la página de inicio de sesión, en donde se propone un elemento tipo *card* con dos botones en la parte superior a modo de pestañas, esto para intercambiar entre la pantalla de ingreso y el formulario de registro para el caso de que el usuario desee registrar una cuenta.

The wireframe illustrates a login screen. At the top left is a small square icon containing a cross. To its right is the text "Logo LANIA". Below this is a horizontal navigation bar with two tabs: "Ingresar" on the left and "¿No tienes cuenta? - Registro" on the right. The main content area is a card with a thin gray border. Inside, the word "Ingresar" is centered above a horizontal line. Below this line is the label "Dirección de correo:" followed by an empty input field. Underneath the input field is the label "Contraseña:" followed by another empty input field. To the right of the "Contraseña:" label is a blue link "Olvidé mi contraseña". At the bottom of the card is a large blue button with the word "Ingresar" in white.

Figura 17. Pantalla de inicio de sesión.

En la siguiente figura se muestra el contenido del formulario propuesto para el registro del aspirante dentro del sistema SIRA:

Logo LANIA

Ingresar Registro

Registro de aspirante:

Nombre: _____ Apellidos: _____

Escuela de procedencia: _____ Número de whatsapp: _____

Correo electrónico: _____

Contraseña: _____ Confirmar contraseña: _____

He leído los términos y condiciones

Registrarse

Figura 18. Pantalla de registro de usuario aspirante.

Para el momento en el que el usuario confirma su cuenta desde el correo electrónico, se propone una pantalla sencilla en donde se muestra el mensaje: “Su cuenta ha sido confirmada con éxito”, acompañada de un icono de palomita y un botón en la parte inferior que al darle clic redireccione hacia la página de inicio de sesión de SIRA. Dicha pantalla se presenta en el wireframe siguiente:



Figura 19. Wireframe de confirmación de cuenta del usuario.

Para el *layout* general de la aplicación, se propuso un diseño responsive con una barra en la parte superior de la pantalla, en la cual se despliega el título de la aplicación y del lado derecho se propuso tentativamente un botón para opciones relacionadas a la configuración de la aplicación o para realizar el cierre de la sesión del usuario.

En dicho *layout*, se propuso un menú desplegable del lado izquierdo el cual se acciona con un botón localizado en la parte superior izquierda y donde se despliegan las opciones a las que tendrá acceso el usuario según el rol al que pertenezca. La idea es que este menú sea mantenido por default de manera desplegada para las versiones de escritorio, y aparece colapsado en las versiones para dispositivos móviles.

De manera general la parte central de la pantalla se aprovecha para mostrar las vistas de contenido de cada una de las opciones a donde ingrese el usuario.

En la siguiente figura se aprecia el *wireframe* diseñado para el *layout* general de SIRA:

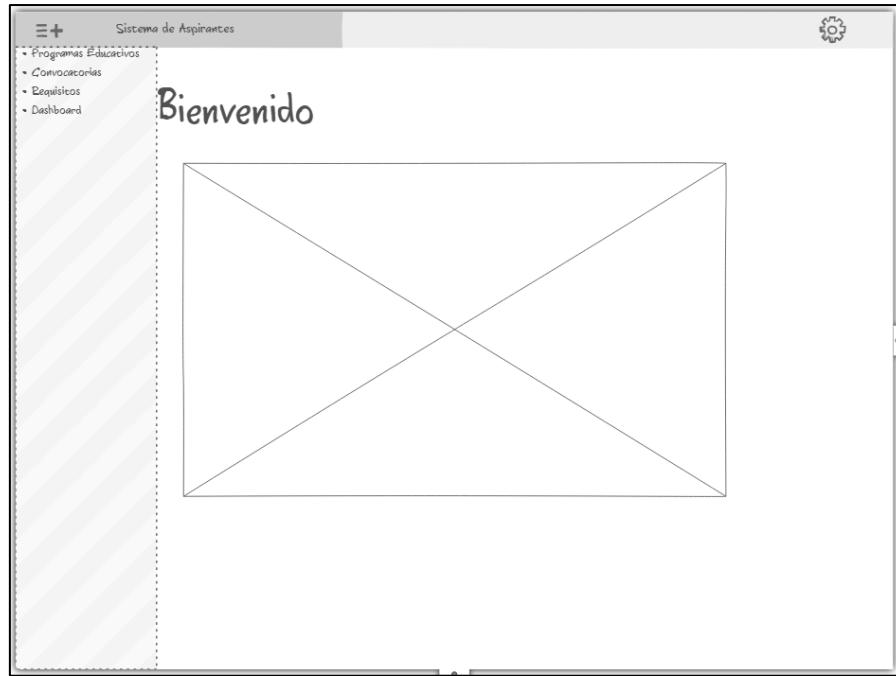


Figura 20. Wireframe de layout general de la aplicación.

Para las pantallas de tipo CRUD, se propuso un estilo tipo tabla en donde se lista el contenido actual del catálogo que está siendo consultado, contando con una columna al lado derecho donde se tengan diferentes iconos de acción para modificar, eliminar o realizar alguna otra acción respecto a cada registro de la tabla.

En la figura siguiente se muestra la pantalla genérica del *wireframe* propuesto para los CRUD de los catálogos de SIRA:

Figura 21. Wireframe general para CRUD de catálogos.

En la parte superior derecha de la tabla CRUD, se cuenta con un botón para crear un nuevo registro, el cual al dar clic aparece un diálogo en donde se tiene el formulario con los datos necesarios de entrada para guardar el nuevo registro.

Se propone el uso de componentes de tipo calendario para seleccionar las fechas y de tipo *number* para ingresar números, y se tome en cuenta este tipo de componentes para dispositivos móviles.

En la figura siguiente se muestra el wireframe de formulario para ingresar un nuevo registro de convocatoria:

Nueva Convocatoria

Nombre

Descripción

Fecha Inicio

Fecha Término

Grupo

Figura 22. Wireframe de formulario de nuevo registro para CRUD.

Para el seguimiento de las participaciones del lado del aspirante, se propuso una vista donde en la parte superior se despliega un tipo *card*, describiendo los datos generales de la convocatoria en la que está participando el aspirante, y en la parte inferior una tabla donde se liste cada uno de los requisitos solicitados por la convocatoria y con botones de acción para subir o visualizar los archivos de cada requisito.

En la siguiente figura se muestra el *wireframe* del seguimiento de participación del aspirante:

Participación de Convocatoria MRYSI 2021

Estatus de la participación											
Entregados 1 de 3											
<table border="1"> <thead> <tr> <th>Requisito</th> <th>Entregado</th> <th>Subir archivo</th> </tr> </thead> <tbody> <tr> <td>Requisito 1</td> <td>SI</td> <td>Seleccionar archivo</td> </tr> <tr> <td>requisito 2</td> <td>NO</td> <td>Seleccionar archivo</td> </tr> </tbody> </table>			Requisito	Entregado	Subir archivo	Requisito 1	SI	Seleccionar archivo	requisito 2	NO	Seleccionar archivo
Requisito	Entregado	Subir archivo									
Requisito 1	SI	Seleccionar archivo									
requisito 2	NO	Seleccionar archivo									

Figura 23. Wireframe de participación del aspirante.

En cuanto a la funcionalidad de tablero o *dashboard* de control, se tomó en cuenta el concepto de “*Análisis de Drill-down*” [21], en donde se va mostrando la información de lo general a lo particular, para lo cual se propuso un tablero estadístico que despliega en la parte superior de manera clara y vistosa tres datos importantes que son: número de aspirantes registrados, número de convocatorias activas y programas educativos activos. Finalmente, en la parte inferior se listan por convocatoria activa, los datos generales como fecha de inicio y término, el cupo y un botón de acción para visualizar los aspirantes de la cada una de las convocatorias listadas.

En la siguiente figura se muestra el *wireframe* para el dashboard propuesto con los elementos visuales mencionados de manera general:

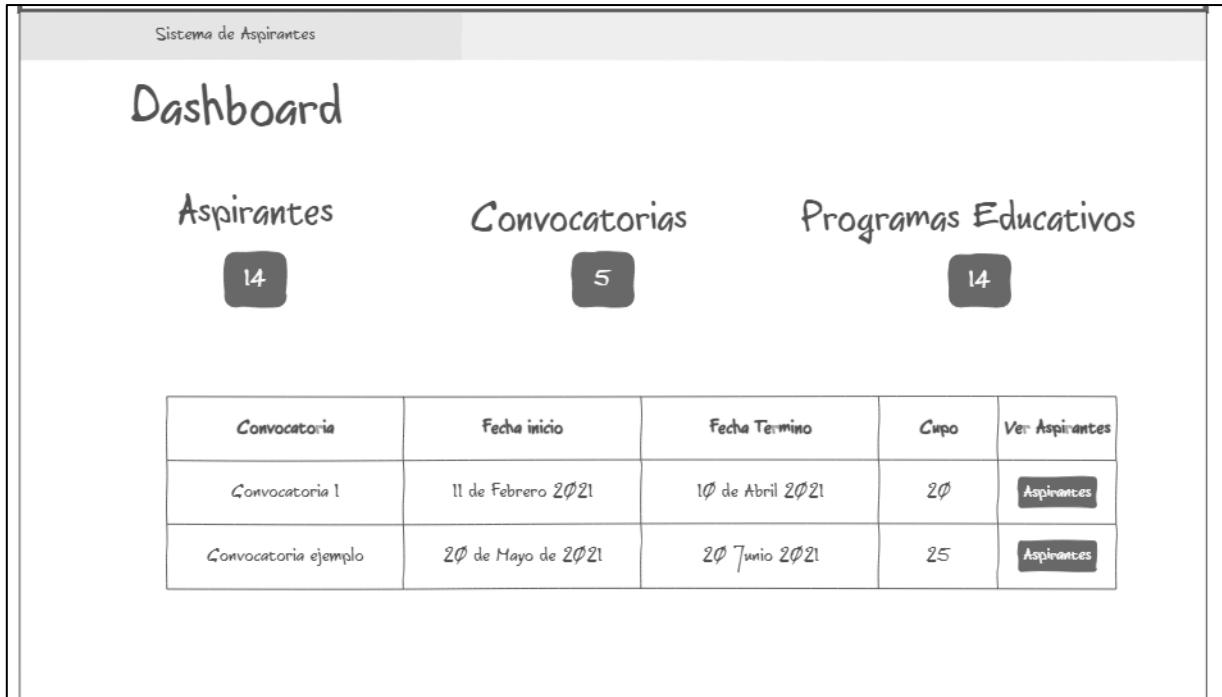


Figura 24. Wireframe de diseño de dashboard

La idea del enfoque propuesto, es permitir presentar los datos de manera general en primera instancia, y a través de los botones de acción ir hacia el detalle si se desea, como por ejemplo consultar el estatus de una convocatoria en específico, observando el listado de todos los aspirantes registrados en ella y su estatus; y de igual manera en la siguiente vista poder ir a más detalle, permitiendo consultar el estatus de la participación de manera específica por cada aspirante que se deseé.

4.2.2 Estructura de los componentes con el framework VuetifyJS.

Una de las características importantes del uso de *frameworks* como *VuetifyJS*, es que permite disgregar la arquitectura de la aplicación en distintos componentes, donde a partir de un componente principal o raíz, se van incrustando los componentes necesarios.

Dentro de cada componente se definieron ciertos objetos JavaScript que fungen como modelos de datos, los cuales son actualizados de manera dinámica a partir de peticiones asíncronas que se realizan hacia los servicios REST del Back-end, (esto mediante la herramienta AXIOS de JavaScript), y permitiendo así a través de la tecnología de Vue, renderizar los componentes visuales sin tener que recargar la página entera desde el servidor.

Así mismo se realizó el uso de la librería *RouterLink* de *VueJs*, que permite renderizar componentes dentro de otros, en base a las rutas definidas dentro de un archivo de configuración *router/index.js*.

Dentro del proyecto *VuetifyJs*, se definieron los componentes de la siguiente manera:

- **Componente de Layout General:** Dentro de este componente se implementó la vista de layout general, consta de una barra de navegación en la parte superior, y del lado izquierdo un menú colapsable donde se despliegan las distintas opciones a las que se puede acceder donde dependiendo del tipo de usuario se desplegaran distintas opciones disponibles.
- **Componente de Acceso:** Este componente engloba el formulario de acceso al sistema.
- **Componente de Registro:** Este componente engloba el formulario de registro para el aspirante.

- **Componentes tipo CRUD:** De manera general para todos los CRUD del sistema SIRA, se implementó un componente tipo CRUD que incluye un listado a modo de tabla para desplegar los datos, donde se incluye una columna con botones de acción para editar o eliminar cada registro de la tabla, y además se incluye en la parte superior un botón que despliega formulario para ingresar un nuevo registro. Este componente se utiliza para Convocatorias, Programas Educativos, Requisitos y Usuarios.
- **Componente Convocatorias:** Este componente es un listado de las convocatorias activas disponibles, las cuales se muestran a los aspirantes en el apartado de “convocatorias”.
- **Componente Participaciones Aspirante:** Este componente es básicamente un listado a modo de tarjetas o *cards*, donde se despliega información general de las convocatorias en las que participa un aspirante.
- **Componente Participación Aspirante:** En este componente, se muestra en la parte superior información general sobre la convocatoria a la que está participando el aspirante, así como el listado de requisitos a modo de tabla, donde se cuenta con botones de acción para subir los documentos del aspirante.
- **Componente Seguimiento Participación:** En este componente, se muestra en la parte superior información general sobre la convocatoria a la que está participando el aspirante, así como el listado de requisitos a modo de tabla, el estatus que tienen de entregado y botones de acción para visualizar y validar cada documento.
- **Componente Seguimiento Convocatoria:** En este componente, se muestra en la parte superior información general sobre la convocatoria a la que están participando los aspirantes, así como el listado de aspirantes a modo de tabla, y botones de acción para visualizar a detalle cada participación.
- **Componente Dashboard:** En este componente, se muestran algunas estadísticas a modo de tablero como son: número de aspirantes

registrados, número de convocatorias activas, programas educativos y el listado de convocatorias para consultar las participaciones.

En la siguiente figura se muestra lo descrito anteriormente:

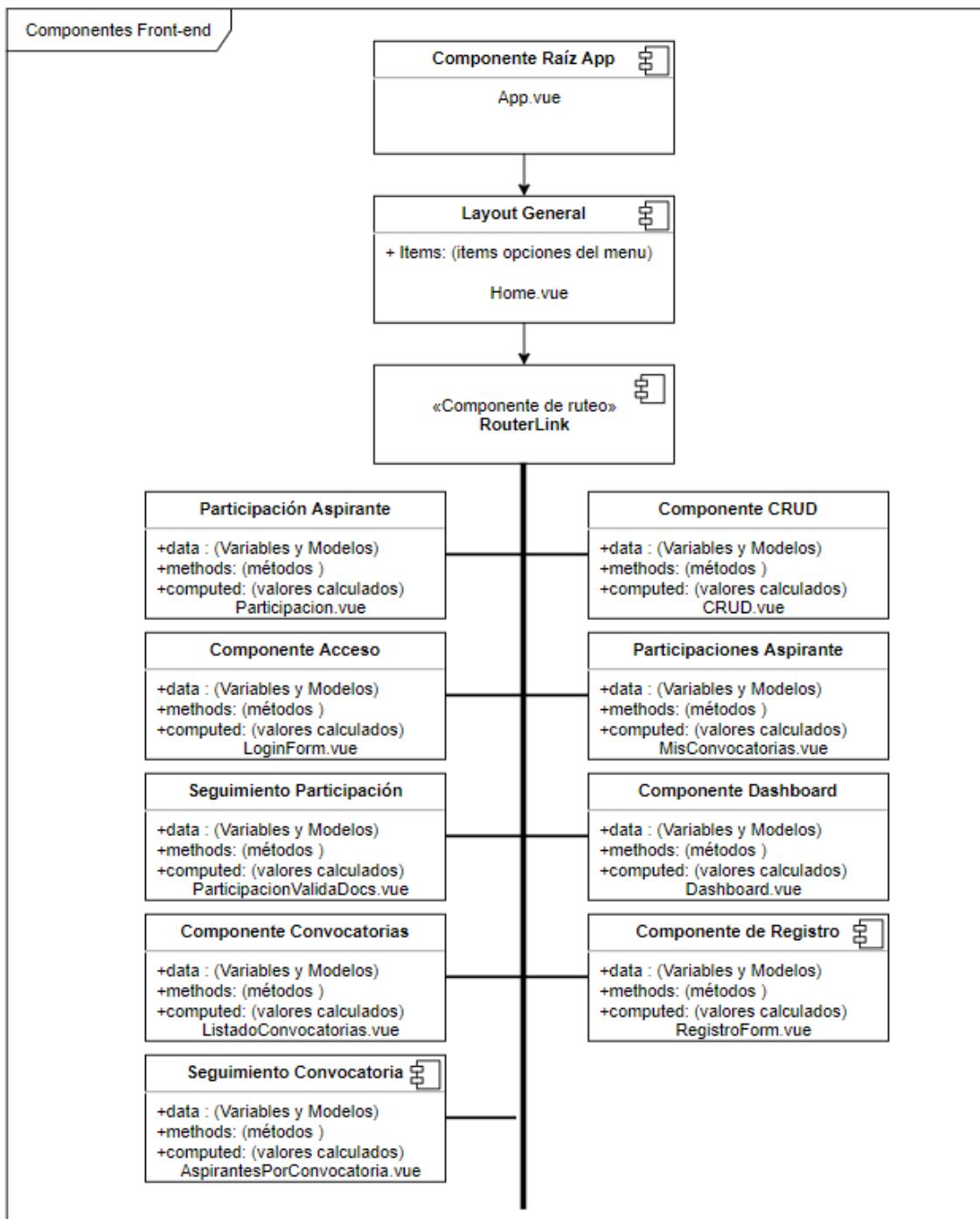


Figura 25. Diagrama de Componentes del Frontend.

5. Descripción del prototipo funcional implementado.

De acuerdo al análisis descrito en el capítulo 2, así como en el diseño de los servicios y el diseño de la interfaz propuesta (capítulos 2 y 3), se realizó como objetivo final la implementación de un prototipo funcional del sistema SIRA. Dicho prototipo se separó en dos proyectos:

- 1) El cliente WEB con la interfaz responsive, el cual fue realizado en el lenguaje JavaScript con el *framework VuetifyJs* en su versión 2.4.
- 2) El proyecto *backend*, que contiene los servicios REST, el cual se desarrolló en el lenguaje Java en su versión OpenJDK 14, mediante el Framework “SpringBoot” en su versión 2.4.4. Para la base de datos se utilizó PostgreSQL en su versión 13.

Para mayor referencia observar el diagrama de Arquitectura de SIRA del capítulo 3 (Figura 7).

5.1 Instalación y ejecución del cliente WEB.

Para poder ejecutar e instalar el cliente WEB se debe tener instalado el manejador de paquetes de JavaScript “npm”.

Para instalar las dependencias se debe correr desde una terminal el comando:

```
# npm install
```

Para compilar y ejecutar el proyecto en modo desarrollo, ejecutar el comando:

```
# npm run serve
```

Para compilar y ejecutar el proyecto para modo producción, correr el comando:

```
# npm run build
```

El proyecto WEB se desarrolló utilizando la herramienta “*vue-cli*”, donde se definió agregar las librerías de “*vue-router*” y “*vuex*”, con la que se genera una

estructura definida para proyectos *VuetifyJS*. (Para mayor referencia consultar la documentación de *VuetifyJs* y *VueJs* [17] y [16]).

Es importante señalar que para las peticiones hacia los servicios REST, se encuentra configurada una ruta base para usar con la librería AXIOS. Dicha configuración se encuentra dentro del archivo *src/main.js*, en donde se definió como:

```
axios.defaults.baseURL = 'http://localhost:8081/';
```

Esto tomando en cuenta que los servicios REST se encuentren en el mismo host y en el puerto 8081, tal como se configuró en el prototipo; la cual puede ser modificada si así se desea.

5.2 Instalación y ejecución del proyecto Backend.

El proyecto Java *Spring Boot*, se generó haciendo uso del manejador de dependencias *Maven*, por lo que en primer lugar se deben instalar las dependencias del proyecto una vez teniendo *Maven* instalado, ejecutando el comando:

```
# mvn clean install
```

Una de las características de los proyectos de Java *Spring Boot* utilizando las librerías para proyectos tipo WEB, es que ya incluyen embebido un servidor de aplicaciones *Tomcat*, el cual es invocado al ejecutarse como un proyecto de tipo *Maven*, esto puede ser realizado a través del mismo IDE de desarrollo como *Eclipse*, *NetBeans*, *IntelliJ*, etc., o simplemente desde la terminal de comandos ejecutándose como un archivo empaquetado *.jar* con:

```
# java -jar demo-0.0.1-SNAPSHOT.jar
```

Por default dentro del prototipo se configuró como puerto de escucha el puerto 8081, así como la conexión a la base de datos de la siguiente manera:

```
datasource:  
    password: 123456  
    url: jdbc:postgresql://localhost:5432/sira  
    username: postgres
```

Dichos parámetros se pueden configurar dentro del proyecto *Java Spring Boot*, modificando el archivo de configuración general de tipo *yml* ubicado en el directorio:

src/main/resources/application.yml

Dentro de dicho archivo también se debe configurar el acceso al servidor de correos, para poder enviar las notificaciones.

Para mayor referencia consultar la documentación de *Java SpringBoot Framework* [20]).

5.3 Descripción de funcionalidades del prototipo.

A continuación, se describen las diferentes pantallas y funcionalidades del prototipo desarrollado divididas por tipo de usuario o rol: *aspirante*, *administrador* y *seguimiento*.

5.3.1 Registro de usuario aspirante e ingreso al sistema.

Se trata de la pantalla de bienvenida del sistema. Conformada por una tarjeta con dos opciones en la parte superior a modo de pestañas, tanto para iniciar sesión en el sistema, como para registrarse en el caso de ser un usuario de tipo aspirante.

Para el caso de la pantalla de registro, se muestra el formulario donde el aspirante procede a llenar datos cómo: nombre, apellidos, escuela de procedencia, número de *WhatsApp*, correo y contraseña la cual debe tener por lo menos 6 caracteres de longitud; los campos escuela de procedencia y número de *WhatsApp* no son obligatorios. Así también se muestra el botón para consultar el “aviso de privacidad”.

En la siguiente figura se muestra la pantalla tipo diálogo donde se muestra el formulario de registro del aspirante:

The screenshot shows the registration page for LANIA. At the top, there are two buttons: "INGRESAR" (Login) with a user icon and "REGISTRARSE" (Register) with a user icon. The main title "Registro" is centered above the input fields. The form consists of several input fields with placeholder text and validation icons:

- Nombre: Miguel Ángel
- Apellidos: Landa López
- Escuela de procedencia: Universidad Veracruzana
- No. de whatsapp: 2299334421
- Correo: mlanda.mrysi19@lania.edu.mx
- Contraseña: 123456 (with a visibility icon and a character count of 6)
- Confirmar contraseña: 123456 (with a visibility icon and a character count of 6)

A large green "REGISTRAR" button is at the bottom. Below it, a blue box contains the link "AVISO DE PRIVACIDAD". A note below the link states: "Al crear su cuenta usted acepta el acuerdo de privacidad de LANIA."

Figura 26. Pantalla de registro del aspirante.

Una vez que el usuario da clic en el botón “REGISTRAR”, se desplegará un mensaje de registro exitoso, y le será enviado un correo con una liga, conformada por un *token* de confirmación para que el aspirante confirme su cuenta al hacer clic en dicha liga.

La pantalla del correo de confirmación se muestra en la siguiente figura:

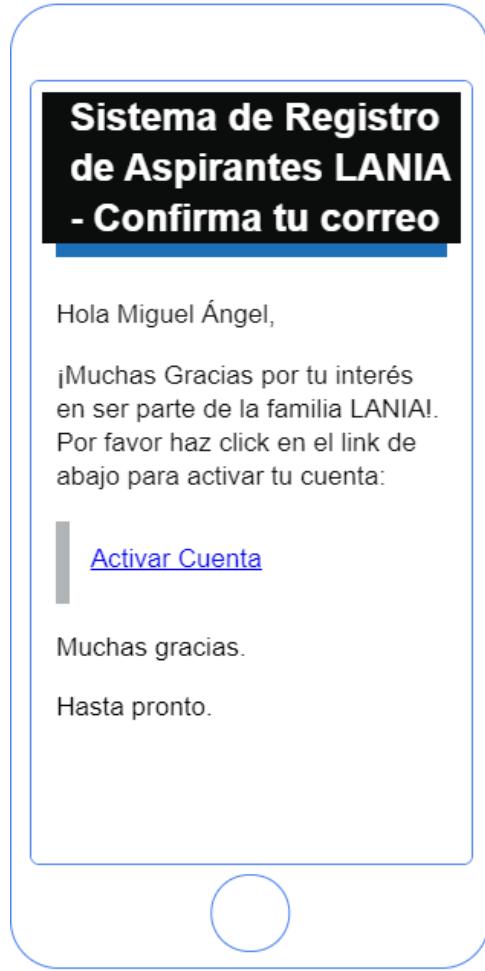


Figura 27. Pantalla de correo de confirmación de cuenta de aspirante.

Una vez que el usuario aspirante confirma su cuenta, será redirigido hacia una pantalla con mensaje de éxito, donde puede hacer clic a la liga de acceso a SIRA:

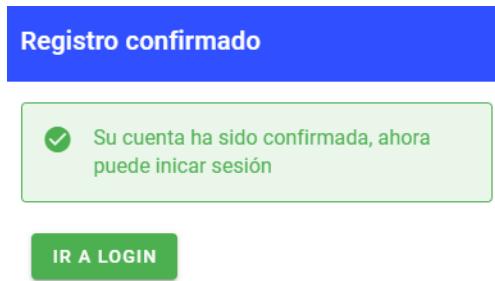


Figura 28. Pantalla de confirmación de cuenta exitosa.

En la figura siguiente se presentan las pantallas de inicio y registro en su resolución para dispositivos móviles:

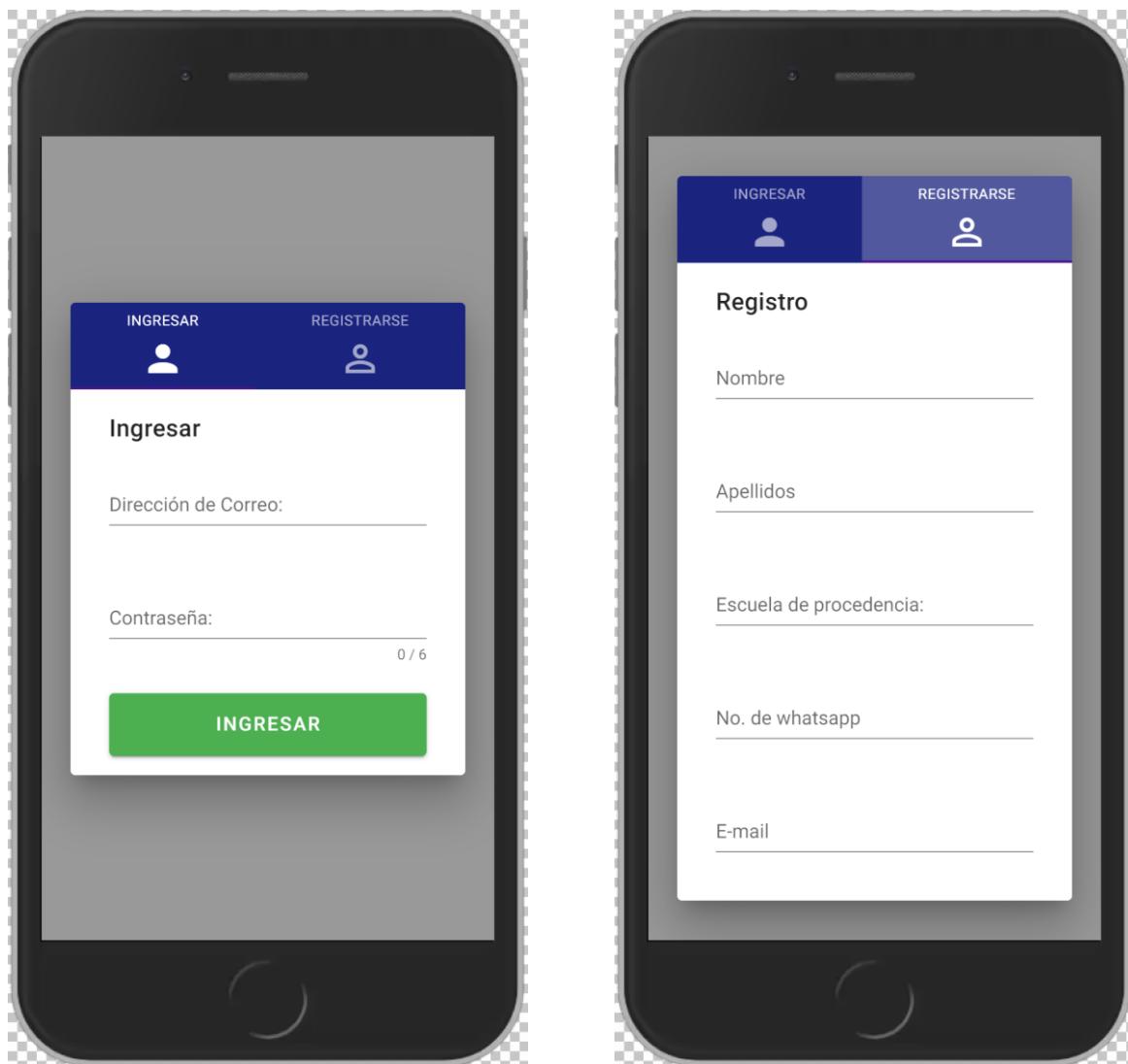


Figura 29. Pantallas de ingreso y registro en resolución móvil.

Una vez que el usuario ingresa sus credenciales correctas dentro de la pantalla de inicio de sesión y hace clic en el botón “INGRESAR”, será redirigido a su pantalla principal de SIRA dependiendo del tipo de rol del usuario.

Para los roles de Administrador y Seguimiento se redirecciona a la pantalla de *Dashboard*, mientras que, para el rol Aspirante, se redirige hacia la pantalla de convocatorias disponibles.

5.3.2 Funcionalidades del rol aspirante.

Una vez que el usuario aspirante ingresa de manera exitosa al sistema, se muestra la pantalla de “Convocatorias Disponibles”, donde se muestra a manera de listado las convocatorias en las que puede participar.

En la esquina superior izquierda, el aspirante puede hacer clic para desplegar el menú general de la aplicación, contando con las opciones de “Convocatorias Disponibles” y “Mis participaciones”:



Figura 30. Opciones del menú del aspirante.

Dentro de la vista “Convocatorias Disponibles”, se mostrará una tarjeta por cada convocatoria disponible, donde cada tarjeta de convocatoria cuenta con un botón “VER DETALLE”, para consultar los requisitos y detalles de la convocatoria seleccionada.

En la siguiente figura se muestra la pantalla de “Convocatorias Disponibles” del usuario aspirante:



Figura 31. Pantalla de convocatorias disponibles del aspirante.

Una vez que se selecciona alguna para ver el detalle, se muestra un diálogo con datos importantes como fechas de la convocatoria, listado de requisitos, el botón para poder participar en la misma y el botón de cierre del diálogo.

En la figura siguiente se muestra el diálogo de detalle en donde el usuario aspirante puede hacer clic en el botón “PARTICIPAR”, acción que desencadena el registro de la participación del aspirante, y al mismo tiempo el envío de un correo electrónico con los detalles de la convocatoria en la cual se registró la participación.

**Maestría en Redes y Sistemas
Integrados Presencial**
Convocatoria MRySI 2021 - presencial



Fecha Inicio:

2021-07-10

Fecha Término:

2021-07-01

Fecha Examen:

2021-07-18

Hora Exámen:

5:00 pm

Requisitos

Titulo Licenciatura

Titulo de Licenciatura

Certificado de Licenciatura

Certificado de Estudios de Licenciatura

Acta de nacimiento original

Acta de nacimiento

CERRAR

PARTICIPAR

Figura 32. Pantalla de detalle de la convocatoria.

A continuación, se muestra la pantalla una vez que el usuario aspirante hizo clic en el botón participar:

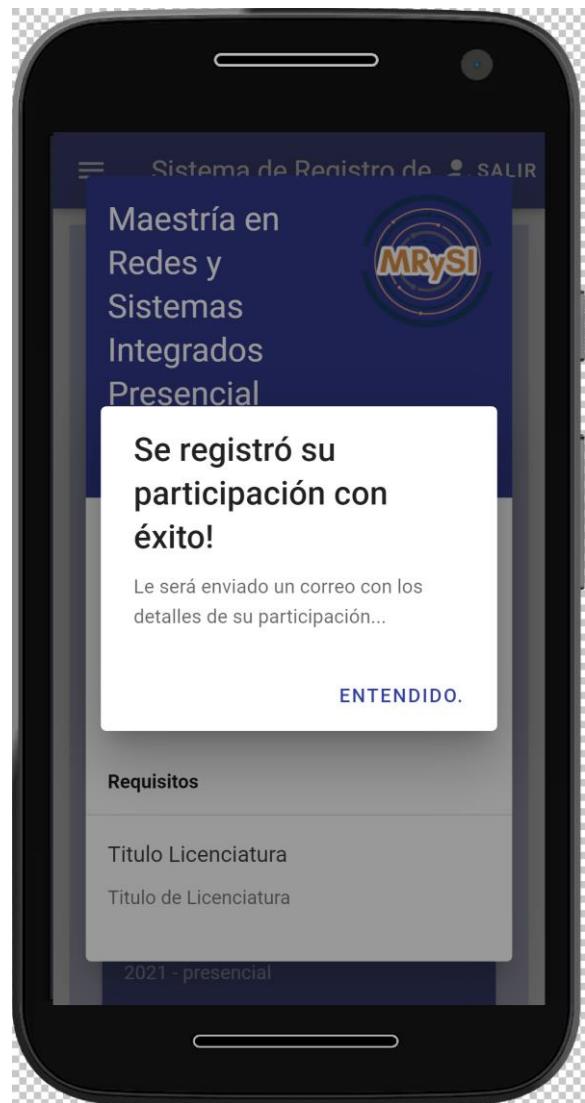


Figura 33. Registro exitoso a convocatoria en resolución móvil.

En la siguiente figura se muestra la pantalla donde se le envía un correo al aspirante con los detalles de la convocatoria a la cual se ha registrado:

Sistema de Registro de Aspirantes LANIA

- Registro de convocatoria exitoso

Hola Miguel,

Tu participación en la convocatoria de Convocatoria MRYSI 2021 - presencial ha sido registrada. Estos son los requisitos que deberás subir al sistema en el apartado de Mis convocatorias

- Titulo Licenciatura
- Certificado de Licenciatura
- Acta de nacimiento original

[Ingresar a SIRA](#)

En cuanto completes tu documentación, ésta será revisada por el personal de LANIA para su validación. Muchas gracias.

Hasta pronto.

Figura 34. Pantalla de correo de registro a convocatoria exitoso.

Dentro de la vista “Mis Participaciones”, el usuario aspirante puede consultar las participaciones a las cuales se ha registrado, de igual manera en forma de pequeñas tarjetas donde aparecen datos como: nombre de la convocatoria, el estatus, la fecha límite de subir documentos y un botón de “VER DETALLE”.

Si el aspirante estuviera participando en más de una convocatoria se muestran todas sus participaciones a modo de listado (mostrando una tarjeta para cada participación).

Mis participaciones



Figura 35. Vista “Mis Participaciones” del aspirante.

Al hacer clic en el botón “VER DETALLE”, se despliega la pantalla de detalle de la participación, donde se muestran los datos generales de la convocatoria, así como los requisitos solicitados a manera de tabla. En cada uno de los requisitos de la participación se tiene la columna de acciones, donde el aspirante puede subir el archivo correspondiente al requisito seleccionado. El tipo de archivos permitidos son en formato PDF, y deben ser menores a 2 MB de tamaño.

A continuación, se muestra la vista de la participación del aspirante en modo de resolución para dispositivos de escritorio:

Requisito	Original/copia	Cantidad	Entregado	Subir archivo	Ver Archivo
Titulo Licenciatura	ambos	1	NO		A...
Certificado de Licenciatura	ambos	1	NO		A...
Acta de nacimiento original	ambos	1	NO		A...

Figura 36. Detalle de la participación para el usuario aspirante.

Una vez que el usuario aspirante carga algún archivo PDF dentro de los requisitos de la convocatoria, aparece el ícono de “Ver Archivo”, donde se puede consultar el archivo subido. Así mismo, el estatus de entregado cambia dinámicamente a la palabra “SI” y se despliega de color verde. De igual manera en la etiqueta de la parte superior, se actualiza el número de documentos entregados del total de requeridos en dicha convocatoria.

Cuando se completan todos los archivos requeridos por la convocatoria, la etiqueta de entregados cambia a color verde.

En la figura siguiente se aprecia el cambio de elementos cuando se sube un archivo de los requisitos solicitados.

The screenshot shows a web-based application for a Master's program in Networks and Integrated Systems (Maestría en Redes y Sistemas Integrados Presencial). The application is for the 2021-2021招ocatoría (Convocatoria MRysi 2021 - presencial).

Key elements visible in the interface:

- Estatus de docs. totales entregados:** Shows "Entregados: 1 de 3" (Delivered: 1 of 3) in a red-bordered box, with an arrow pointing to it from the text "Estatus de docs. totales entregados".
- Botones de acción:** A box containing buttons for "Subir archivo" (Upload file) and "Ver Archivo" (View file), with arrows pointing to them from the text "Botones de acción".
- Table:** A table showing requirements (Requisito) and their status (Original/copia, Cantidad, Entregado). One row is highlighted for "Titulo Licenciatura" with "ambos" original/copy and quantity "1". The "Entregado" column contains a green "SI" button, which is also highlighted with a red border. The "Subir archivo" and "Ver Archivo" buttons are also highlighted with a red border.

Figura 37. Detalle de participación con un requisito entregado.

Ya completado todos los requisitos de la convocatoria, dentro de la pantalla de “Mis Participaciones”, el estatus de la participación se cambia a “en validación”, indicando que se encuentra en proceso de validación de documentos por parte del personal de LANIA.

5.3.3 Funcionalidades del rol de seguimiento.

Cuando el usuario de tipo Seguimiento ingresa de manera correcta sus credenciales dentro de la vista de inicio, se redirecciona a la vista de *Dashboard*, la cual se muestra en la siguiente figura:

The screenshot shows the 'Sistema de Registro de Aspirantes' dashboard. At the top, there is a header with a menu icon, the title 'Sistema de Registro de Aspirantes', and a 'SALIR' button. Below the header, the word 'Dashboard' is displayed in a large, bold font. The dashboard features three main cards: 'ASPIRANTES REGISTRADOS' (1), 'CONVOCATORIAS ACTIVAS' (3), and 'PROGRAMAS EDUCATIVOS' (5). Each card has a green icon and a 'VER' button. A callout box points to the 'VER' button under 'PROGRAMAS EDUCATIVOS' with the text 'Botón para ver la participación de cada aspirante'. Below these cards is a section titled 'Convocatorias Activas' containing a table with three rows of data. The table columns are: No., Convocatoria, Fecha de Inicio, Fecha de Término, Cupo, and Ver Aspirantes. The 'Ver Aspirantes' column for the first row is circled in red and has an arrow pointing to it from the callout box. The table also includes pagination controls at the bottom.

No.	Convocatoria	Fecha de Inicio	Fecha de Término	Cupo	Ver Aspirantes
1	Convocatoria MCA 2021	2021-07-08	2021-07-10	20	
2	Convocatoria MRYSI 2021 - presencial	2021-07-10	2021-07-01	23	
3	Convocatoria DIP-Redes 2021 - presencial	2021-07-10	2021-07-01	23	

Figura 38. Vista de Dashboard del usuario seguimiento.

Como se puede apreciar, se muestran algunas estadísticas en la parte superior como: número de aspirantes registrados, número de convocatorias activas, programas

educativos registrados; mientras que, en la parte inferior, aparece el listado de convocatorias activas con algunos datos generales y el botón de acción: “Ver Aspirantes”, para poder consultar a detalle el listado de aspirantes de cada convocatoria.

Dentro del menú principal, el usuario de seguimiento solo tiene acceso a la opción de *Dashboard*, esto como parte de la implementación del prototipo funcional, sin embargo, podría implementarse también el acceso al listado completo de aspirantes, así como una vista distinta y más detallada de las convocatorias activas en una versión productiva.

Para verificar el estatus de las convocatorias, se debe hacer clic sobre el botón de “Ver Aspirantes” en la tabla de convocatorias activas; de esta manera se redirige hacia la vista de seguimiento de la convocatoria mostrada en la siguiente figura:

The screenshot displays the 'Programa de MCA' interface for the 'Test 2021'招考. At the top, there's a header with the title 'Programa de MCA' and 'Participantes de la Convocatoria Test 2021'. To the right is a logo for 'ADMINISTRACIÓN DE REDES' featuring three stacked servers. Below the header, a red box highlights the 'Fechas importantes' (Important Dates) section, which includes the 'Fecha Inicio' (20/02/2021), 'Fecha Término' (24/03/2021), 'Fecha Examen' (28/03/2021), and 'Hora Exámen' (5:00 pm). A callout box labeled 'Fechas importantes' points to this section. Another callout box labeled 'Botones de acción' points to the 'subir requisitos' (Upload Requirements) button in the applicant table. Arrows also point from the 'Estatus de la participación' (Participation Status) box at the bottom left to the 'Estatus' column in the table, and from the 'REGRESAR' (Return) button at the bottom right to the table itself. The table lists two applicants: Pablo Ramírez González and Juana Díaz, both from 'Uv'. Each applicant row has a 'subir requisitos' button and a circular icon with a checkmark. The bottom of the screen shows pagination controls: 'Rows per page: 10', '1-2 of 2', and navigation arrows.

Aspirante	Escuela Procedencia	Estatus	Ver documentos
Pablo Ramírez González	Uv	subir requisitos	
Juana Díaz	Uv	subir requisitos	

Figura 39. Pantalla de seguimiento de convocatoria

Se puede observar que se muestran los datos generales de la convocatoria, así como fechas importantes en la parte superior. En la parte inferior se encuentra el listado de aspirantes a dicha convocatoria, así como el estatus de su participación. Se cuenta también con un botón de acción de “Ver Documentos”, con el cual el usuario puede acceder a la vista detallada del estatus de la participación de un aspirante en específico.

Al hacer clic dentro de “Ver Documentos”, se redirige a la vista de “Estatus de la participación del Aspirante”, donde se puede apreciar el estatus de los documentos entregados y validados. En la parte superior se visualizan los datos generales de la participación como: nombre del aspirante, escuela de procedencia, número de WhatsApp, la hora del examen; mientras que en la parte inferior a manera de tabla se listan los requisitos de la convocatoria, y dos botones de acción: “Ver Doc.”, para visualizar el documento subido por el aspirante, y “Validar Doc.”, para registrar como válido el documento subido por el aspirante por parte del usuario seguimiento.

Requisito	Original/copia	Cantidad	Indispensable	Ve Doc.	Validar Doc.
Titulo Licenciatura	ambos	2	SI	<input checked="" type="radio"/>	VALIDAR
Certificado de Licenciatura	ambos	2	SI	<input checked="" type="radio"/>	OK
Acta de nacimiento original	original	1	SI	No entregado	VALIDAR

Figura 40. Pantalla de seguimiento a la participación del aspirante.

Una vez que se validan todos los documentos subidos por el aspirante, aparece un mensaje de que se ha completado la validación, y el estatus cambia a color verde indicando que se ha validado todos los documentos por parte del personal de LANIA.

Es importante señalar que para todas las vistas que manejan tablas, se consideró una disposición de elementos de manera vertical, para facilitar la lectura en dichos dispositivos, tal como se muestra en la siguiente figura:

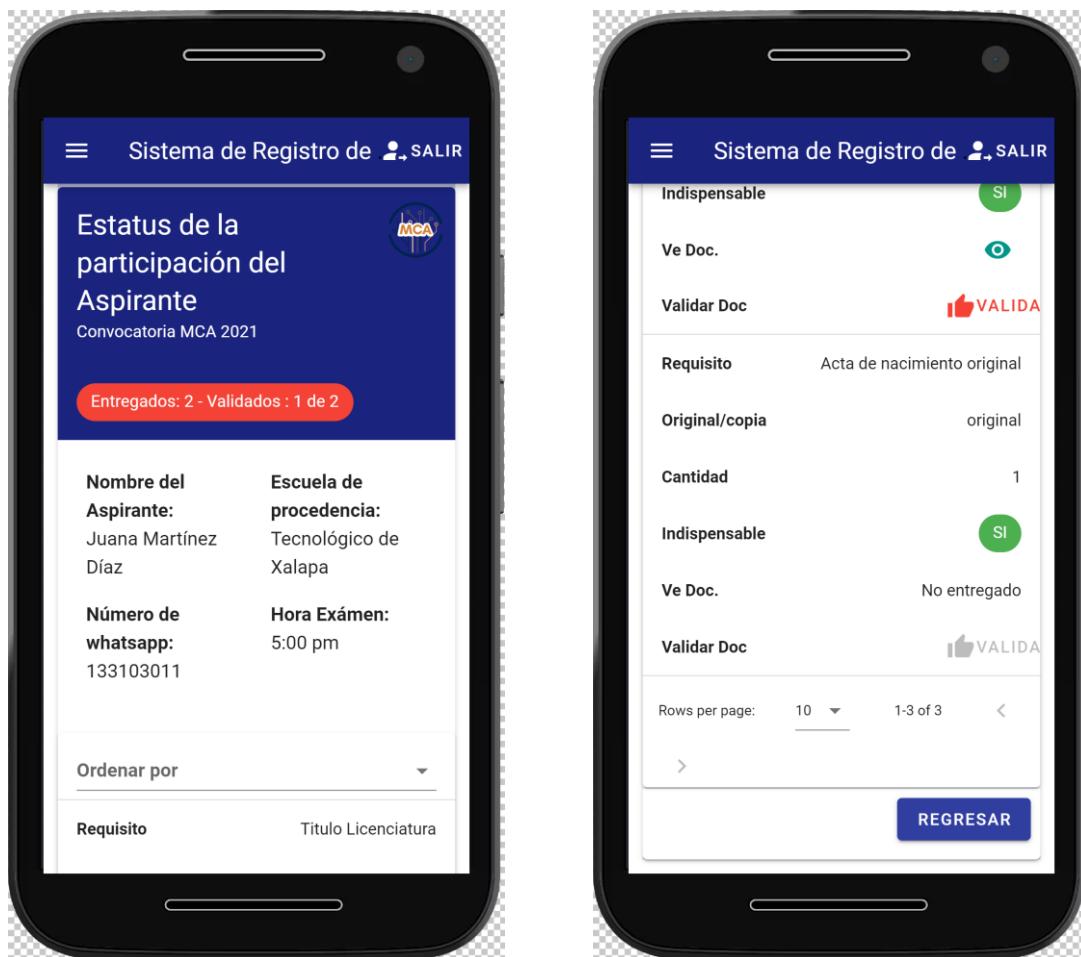


Figura 41. Estatus de la participación del aspirante (resolución móvil).

5.3.4 Funcionalidades del rol administrador.

Para poder ingresar como usuario administrador, se creó por defecto el usuario `admin@lania.edu.mx` con contraseña 123456, sin embargo, se pueden dar de alta cuantos administradores se necesiten desde el CRUD de usuarios del sistema.

Una vez ingresadas las credenciales como usuario administrador dentro de SIRA, se accederá a la vista de *dashboard*, la cual cuenta con el mismo aspecto que la del usuario de tipo seguimiento.

Dentro del menú desplegable de la aplicación en la parte superior izquierda, se listan todas las opciones que se pueden acceder con el usuario administrador:



Figura 42. Menú desplegable del usuario administrador.

La vista de *dashboard* cuenta con las mismas opciones que tiene el usuario de seguimiento, es decir, que el usuario administrador también puede verificar los documentos y realizar el seguimiento del estatus de las participaciones de los aspirantes.

Las demás opciones a las que puede acceder el administrador se trata de las funcionalidades de alta, baja y modificaciones de: convocatorias, programas educativos, requisitos de convocatorias y usuarios del CEL.

Dentro de la vista de catálogos de convocatorias, se muestra a manera de listado en una tabla, todas las convocatorias dadas de alta en el sistema SIRA, teniendo en

la parte superior derecha un botón para agregar una nueva convocatoria, así como botones de acción en la columna “acciones”, para poder modificar o eliminar alguna existente, o bien, visualizar el diálogo para agregar o eliminar requisitos a alguna convocatoria:

Convocatorias		Botón de creación de nueva entidad						
No.	Nombre	Descripción ↑	Fecha Inicio	Fecha Término	Fecha Examen	Programa Educativo	Cupo	Acciones
1	Convocatoria MCA 2021	Convocatoria de la Maestría en Computación Aplicada 2021	7 jul. 2021	7 jul. 2021	14 jul. 2021	Maestría en Computación Aplicada	20	
2	Convocatoria MRYSI 2021 - presencial	Convocatoria de la Maestría en Computación en Redes y Sistemas Integrados Presencial	9 jul. 2021	9 jul. 2021	17 jul. 2021	Maestría en Redes y Sistemas Integrados Presencial	23	
3	Convocatoria DIP-Redes 2021 - presencial	Convocatoria del Diplomado en Administración de Redes 2021	9 jul. 2021	9 jul. 2021	17 jul. 2021	Diplomado en Administración de Redes	23	

Figura 43. CRUD de Convocatorias para el rol administrador.



Figura 44. CRUD de Convocatorias en resolución móvil

Al hacer clic en el botón de “Crear Nueva”, se abre el diálogo con el formulario para crear una convocatoria nueva, donde se deben ingresar datos como: programa educativo, nombre, descripción, fecha de inicio, fecha de término, fecha de examen y cupo. Al guardar se actualiza la tabla de convocatorias dinámicamente con el nuevo registro.

Se puede observar en la figura anterior la disposición de los elementos de manera horizontal en la resolución de dispositivos móviles.

En la siguiente figura se observa el formulario de entrada para una nueva convocatoria:

Nueva Convocatoria

Maestría en Computación Aplicada

Nombre
Convocatoria 2021 MCA

Descripción -
Convocatoria de la Maestría en Computación Aplicada

Fecha de Inicio
2021-09-23

Fecha de Término
2021-10-30

Fecha de Exam...
2021-10-09

Cupo
25

CANCELAR GUARDAR

Figura 45. Formulario de creación de nueva convocatoria.

Para poder editar una convocatoria se debe hacer clic en el botón de acción con el icono de lápiz, donde se muestra el diálogo con el formulario de la convocatoria con los datos actuales, al modificar y presionar en guardar se guardan los cambios a dicha convocatoria.

Para eliminar una convocatoria, se debe hacer clic en el botón de acción con el icono de bote de basura, donde se mostrará un diálogo preguntando si está seguro de proceder con el borrado, en caso de ser así se elimina dicha convocatoria.

Para agregar requisitos a una convocatoria, se debe hacer clic en el botón de la bitácora que se muestra en la imagen. Se despliega un diálogo con el listado de requisitos actuales de la convocatoria, así como un pequeño formulario para agregar un requisito. Es importante señalar que para poder agregar un requisito primero debe

ser dado de alta dentro del catálogo de requisitos en la opción del menú del administrador “Requisitos”.



Figura 46. Botón de acción de requisitos de convocatoria.

En la siguiente figura se muestra el diálogo con el listado de requisitos a agregar o quitar en la convocatoria seleccionada:

A screenshot of a dialog box titled "Requisitos de Convocatoria MRYSI 2021 - presencial". The main content is a table listing three requirements: "Titulo Licenciatura", "Certificado de Licenciatura", and "Acta de nacimiento original". Each row has a green "SI" button, a quantity "1", both "ambos" options selected, and a small trash can icon. Below the table are pagination controls: "Rows per page: 5", "1-3 of 3", and navigation arrows. At the bottom, there's a link "Agrega un nuevo requisito", a dropdown "Selecciona requisito", a dropdown "original/co...", a toggle switch "Es indispensable", a "Cantidad" input field, and a blue "AGREGAR REQUISITO" button.

Figura 47. Diálogo de requisitos de convocatoria.

Dentro de la opción de “Programas Educativos”, al igual que en la vista de convocatorias, se tiene un listado a modo de tabla con los planes o programas

educativos dados de alta, y también se cuenta con dos botones de acción para poder editar y eliminar un programa educativo.

The screenshot shows a table titled "Programas Educativos" with the following columns: No., Nombre, Descripción ↑, Clave, Vigencia, Abreviatura, and Acciones. The "Acciones" column contains two buttons for each row: a green pencil icon for editing and a red trash bin icon for deleting. A red box highlights the "CREAR NUEVO" button at the top right of the table header. A red box also highlights the "Acciones" column. An arrow points from the text "Botones de acción edición y eliminar" to the "Acciones" column.

No.	Nombre	Descripción ↑	Clave	Vigencia	Abreviatura	Acciones
1	Maestría en Computación Aplicada	Adquiere conocimientos avanzados en tecnologías emergentes: Ingeniería de Software, Sistemas Distribuidos, Bases de Datos, Inteligencia artificial y TIC's en la educación.	MCA-2941	1 jul. 2029	MCA	
5	Maestría en Redes y Sistemas Integrados en Línea	Adquiere una visión integral y sólida en el uso de la tecnología para: la construcción y diseño de sistemas computacionales, el manejo de datos y la administración de redes desde la modalidad en Línea a través de las plataformas que te brinda LANIA	MRYSI-L9310	1 ene. 2029	MRYSI-L	
2	Maestría en Redes y Sistemas Integrados Presencial	Adquiere una visión integral y sólida en el uso de la tecnología para: la construcción y diseño de sistemas computacionales, el manejo de datos y la administración de redes.	MRYSI-020MEP	1 may. 2028	MRYSI-P	
3	Diplomado en Administración de Redes	En este diplomado aprenderás todo el proceso de Ruteo configuración y manejo de Redes Lan y WAN	DPA-12001	1 may. 2028	DPA	
4	Diplomado en Ingeniería de Software	En este diplomado aprenderás todo el proceso, metodologías y herramientas que conlleva todo el proceso de la ingeniería de software	DPA-11001	1 jun. 2028	DPA	

Figura 48. CRUD de programas educativos para el administrador.

Se cuenta con el botón “CREAR NUEVO”, en la parte superior derecha de la tabla para mostrar un diálogo donde se visualiza un formulario para ingresar un nuevo programa educativo. Se muestran datos de ingreso como: nombre del programa educativo, clave, descripción, vigencia y abreviatura.



Figura 49. Nuevo prog. educativo, en resolución escritorio y móvil.

Para editar un Programa Educativo, se debe hacer clic en el botón del lápiz de la tabla de Programas Educativos, donde al mostrarse el diálogo con el formulario se pueden editar los datos actuales y al dar clic en “GUARDAR”, se actualiza la información con los nuevos datos.

Para eliminar un Programa Educativo, se debe hacer clic en el botón con el icono del bote de basura. Al hacer clic se despliega un diálogo de confirmación para la eliminación.

Dentro de la opción de “Requisitos” en el menú del administrador, se sigue la misma mecánica para dar de alta, modificar y eliminar requisitos del catálogo, los cuales son utilizados para agregarse a las convocatorias disponibles.

Una vez que el administrador hace clic en la opción “Requisitos”, se despliega la vista a modo de tabla de todos los requisitos dados de alta en el sistema. También se cuenta con una columna de acciones para poder editar o eliminar cada uno de los requisitos.

Requisitos						CREAR NUEVO
No.	Nombre	Tipo	Descripción	Documento	Cambiante	Acciones
3	Acta de nacimiento original	Documento de identidad	Acta de nacimiento	<input checked="" type="checkbox"/>	NO	 
2	Certificado de Licenciatura	Documento probatorio	Certificado de Estudios de Licenciatura	<input checked="" type="checkbox"/>	NO	 
4	CURP	Documento de identidad	Clave Unica de Registro de Poblacion	<input checked="" type="checkbox"/>	NO	 
1	Titulo Licenciatura	Documento probatorio	Titulo de Licenciatura	<input checked="" type="checkbox"/>	NO	 
Rows per page:						10 <input type="button" value="▼"/>
1-4 of 4						 

Figura 50. CRUD de requisitos para el administrador.

Al igual que en los otros CRUD, se cuenta con un botón de “CREAR NUEVO”, para crear un nuevo requisito, donde se muestra un diálogo con el formulario a llenar con los datos del nuevo requisito.

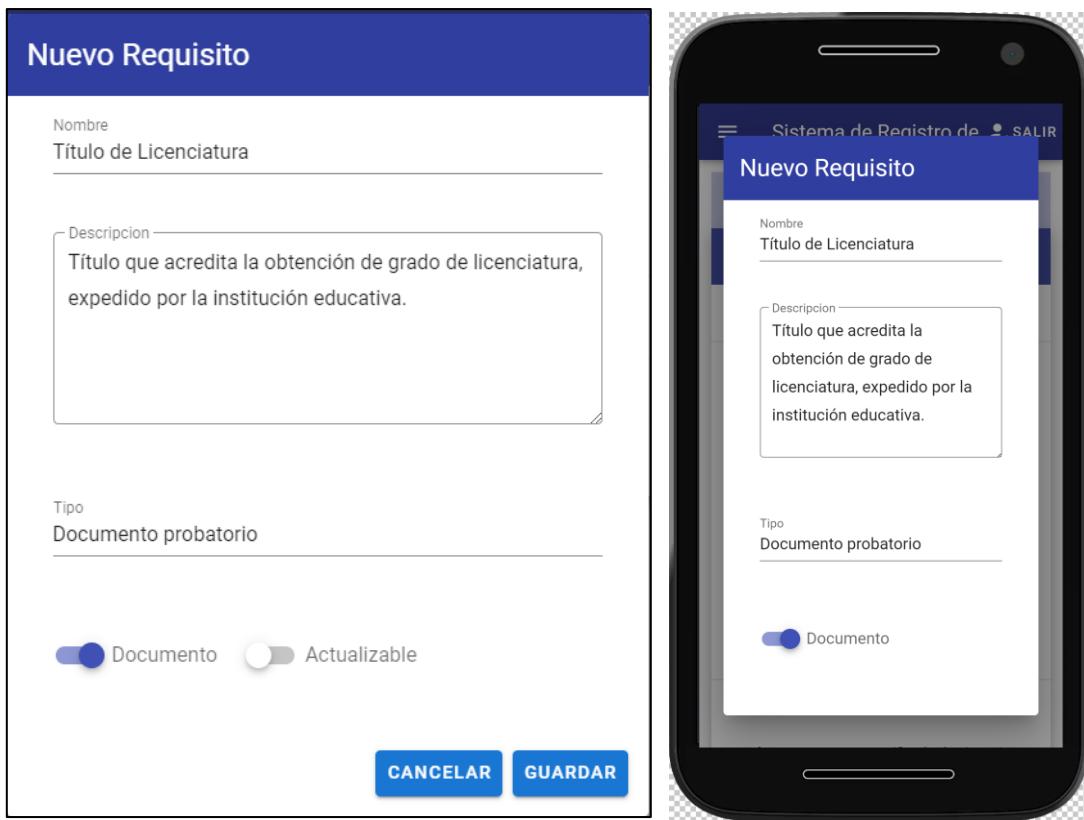


Figura 51. Agregar nuevo requisito, resolución de escritorio y móvil.

Cómo última opción dentro del menú del administrador, se tiene el control de usuarios internos, el cual al darle clic a “Usuarios LANIA” se despliega la vista de usuarios existentes. Dichos usuarios se refieren a los usuarios internos del CEL, es decir los de rol Seguimiento y Administrador.

Al igual que en los anteriores CRUD, se tiene una vista a modo de tabla para poder visualizar todos los usuarios, así como botones de acción para editar y eliminar algún usuario.

Usuarios Lania							CREAR NUEVO
No.	Nombre	Apellido	Email	Clave	Rol	Habilitado	Acciones
1	ADMIN	ADMIN	admin@lania.edu.mx		ADMIN	SI	
2	Raúl	Martínez Godoy	raul@lania.edu.mx	Clave-1234	SEGUIMIENTO	SI	
3	Gustavo	Ruiz	gustavo@lania.edu.mx	gustavo-1223	ADMIN	SI	

Rows per page: 10 < 1-3 of 3 >

Figura 52. *CRUD de usuarios internos para el rol administrador.*

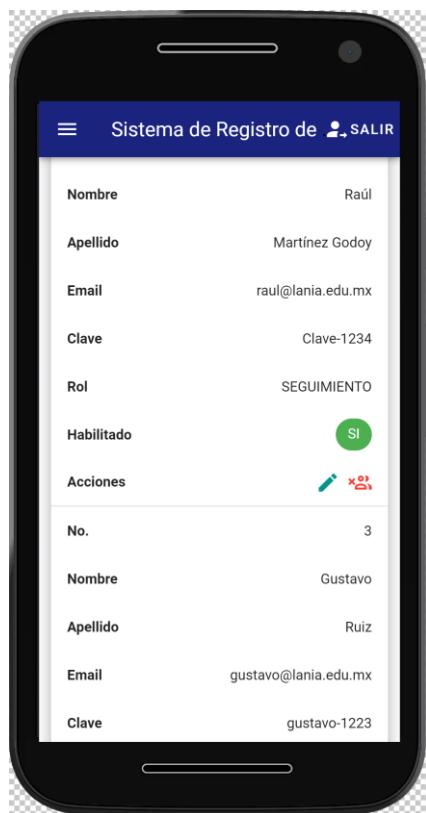


Figura 53. *CRUD de usuarios internos en resolución móvil.*

De la misma manera que los anteriores, se tiene un botón en la parte superior para agregar un nuevo usuario, donde al dar clic se despliega el formulario para ingresar los datos del nuevo usuario.

Editar Usuario

Nombre
Gustavo

Apellido
Ruiz

Correo
gustavo@lania.edu.mx

Password
..... Q
60

Clave de Empleado
gustavo-1223

ADMINISTRADOR ▾

CANCELAR **GUARDAR**

The screenshot shows a modal dialog titled "Editar Usuario". It contains several input fields: "Nombre" (Name) with "Gustavo", "Apellido" (Surname) with "Ruiz", "Correo" (Email) with "gustavo@lania.edu.mx", "Password" (Password) with a masked input, and "Clave de Empleado" (Employee Key) with "gustavo-1223". Below these is a dropdown menu set to "ADMINISTRADOR". At the bottom are two buttons: "CANCELAR" (Cancel) and "GUARDAR" (Save).

Figura 54. Diálogo para nuevo usuario interno del rol admin.

5.3.5 Cerrar sesión del usuario.

Una vez que el usuario ingresa sus credenciales correctas y accede al sistema, se cuenta con un botón de “SALIR”, localizado en la parte superior derecha de la aplicación, donde al hacer clic se redirige al usuario hacia la página de inicio de sesión, y se destruye el *token* de sesión del almacenamiento local del navegador (*localStorage*).



Figura 55. Botón de salir o cerrar sesión.

6. Pruebas y conclusiones.

En este apartado final, se presentan las pruebas funcionales realizadas al prototipo funcional de SIRA, posteriormente se presentan las conclusiones finales del proyecto y trabajos a futuro.

6.1 Casos de prueba.

En las tablas de las siguientes subsecciones, se describen los casos de prueba realizados al prototipo funcional de SIRA, teniendo como objetivo primordial evaluar las funcionalidades disponibles para cada uno de los roles de usuario (Aspirante, Administrador y Seguimiento), y apegándose en todo momento a los casos de uso detallados en el anexo A.

Las pruebas fueron realizadas en un ambiente de pruebas local, utilizando el navegador *Google Chrome* en su versión 97.0.4692.71, así como en *Microsoft Edge* versión 97.0.1072.62 y *Mozilla Firefox* 91.0.2.

Dentro de cada tabla se plasma una fila donde se describen las condiciones de entrada, los valores ingresados, la salida esperada y las condiciones de salida; y si al final dichas condiciones de salida fueron cumplidas, se marcaron con el símbolo “✓” como resultado satisfactorio.

Se entiende por funcionalidad como el conjunto de pasos que satisfacen las necesidades descritas en cada caso de uso. Dichas funcionalidades a evaluar se describen mediante el título de cada tabla comenzando con las siglas CP (caso de prueba) y el nombre de la funcionalidad a evaluar.

6.1.1 Casos de prueba de funcionalidades del aspirante.

Los casos de prueba siguientes presentan el flujo de registro de un aspirante en el sistema SIRA, así como el proceso de su participación a una convocatoria hasta completar los requisitos de la misma.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
1	Nombre: "Miguel Ángel" Apellidos: "Landa López" Escuela de procedencia: "Universidad Veracruzana" Número de WhatsApp: "2288474546" Correo: "mlanda@gmail.com" Contraseña: "123456"	Ingresar un correo electrónico válido	Mensaje de confirmación : "Se ha enviado un correo para confirmar su cuenta"	Se recibe el correo con la liga para confirmar la cuenta del aspirante	✓
2	Liga con el token de confirmación	Aspirante recibe correo de confirmación de su cuenta	Mensaje de confirmación : "Su cuenta ha sido confirmada, ahora puede iniciar sesión"	Se muestra botón para ir a página de inicio de sesión	✓
3	Dirección de correo:"mlanda@gmail.com" Contraseña: "123456"	Haber realizado los escenarios de caso de prueba 1 y 2	Se inicia sesión con los datos de acceso que el aspirante definió	Se muestra la pantalla con el listado de convocatorias para consultar	✓

Tabla 5. CP de registro de usuario aspirante e inicio de sesión.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
4		Usuario aspirante autenticado	Se muestra el listado de convocatorias disponibles	Se despliega botón para visualizar detalle de cada convocatoria.	✓
5	"idConvocatoria". (Al hacer clic en botón "VER DETALLE").	Usuario aspirante autenticado.	Se muestra el listado de requisitos y fechas importantes de convocatoria.	Desplegar botón de "PARTICIPAR".	✓

6	"idConvocatoria" al hacer clic en botón "PARTICIPAR", dentro del detalle de una convocatoria.	Usuario aspirante autenticado.	Mensaje de confirmación: "Se registró su participación con éxito, le será enviado un correo con los detalles de su participación".	Se recibe correo con los detalles de la participación de la convocatoria.	<input checked="" type="checkbox"/>
---	---	--------------------------------	--	---	-------------------------------------

Tabla 6. CP de registro de participación del aspirante a una convocatoria.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
7		Usuario aspirante autenticado. Haber realizado los escenarios de caso de prueba 5 y 6.	Se visualiza la participación del aspirante en la opción "Mis participaciones".	Se despliega pantalla de subir requisitos al hacer clic en el botón "DETALLE" de la participación. Se muestran botones para subir los requisitos. Se muestra botón de regresar a la pantalla anterior.	<input checked="" type="checkbox"/>
8	Archivo PDF	Archivo PDF menor a 2 MB	Se visualiza el nombre del archivo seleccionado	Se visualiza el icono de flecha de color verde para subir el archivo seleccionado	<input checked="" type="checkbox"/>
9	Archivo PDF	Archivo PDF menor a 2 MB	Se visualiza el icono de ojo para visualizar el archivo	Permite visualizar el archivo subido al hacer clic en el icono del ojo. Se incrementa el	<input checked="" type="checkbox"/>

				conteo de requisitos entregados dentro de la etiqueta de estatus .	
--	--	--	--	---	--

Tabla 7. CP de subida de archivos de requisitos de convocatoria a la participación del aspirante.

6.1.2 Casos de prueba de funcionalidades del administrador.

Los casos de prueba siguientes, representan el flujo de inicio de sesión del usuario administrador, así como las altas, bajas y modificaciones de Convocatorias, Programas Educativos, Requisitos y Usuarios internos del CEL.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
10	Dirección de correo:"admin @lania.edu.mx" Contraseña: "123456"	Usuario administrador dado de alta por default.	Se inicia sesión correctamente con las credenciales proporcionadas .	Se despliega el menú colapsable del lado izquierdo para el usuario administrador .	✓

Tabla 8. CP de inicio de sesión del administrador.

Caso de prueba del registro de aspirante:

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
11	Nombre: "Diplomado en Ingeniería de Software, Descripción: "El diplomado te proporciona los conocimientos y competencias requeridas para estructurar el ciclo de vida de un proyecto de ingeniería de software, con un enfoque de calidad a través de la aplicación del paradigma y filosofía ágil", Clave: "DIS-123", Vigencia: "19-09-2021", Abreviatura: "DIS"	Usuario administrador autenticado.	Mensaje de confirmación: "Se guardó correctamente el Programa Educativo"	Se despliega en la tabla el nuevo registro del Programa Educativo ingresado	✓

12	Se modifica el dato de Clave por "DIS-987" y la Vigencia por : "01-08-2029"	Usuario administrador autenticado. Haber realizado el caso de prueba no. 11.	Mensaje de confirmación: "Se guardó correctamente el Programa Educativo"	Se despliega en la tabla el registro modificado del Programa Educativo.	<input checked="" type="checkbox"/>
13	"idProgramaEducativo" al hacer clic en el botón de eliminar.	Usuario administrador autenticado. Haber realizado el caso de prueba no. 11. Haber confirmado la eliminación del P.E. en el diálogo.	Mensaje de confirmación: "Se eliminó el Programa Educativo"	Se actualiza la tabla sin el registro del Programa Educativo eliminado.	<input checked="" type="checkbox"/>

Tabla 9. CP de alta, baja, y modificaciones de programas educativos.

Casos de prueba del CRUD de convocatorias para el rol administrador:

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
14	Programa Educativo: "Maestría en Computación Aplicada", Nombre: "Convocatoria de MCA 2021", Descripción : "Convocatoria de la Maestría en Computación Aplicada en modalidad presencial 2021", Fecha de inicio: "01-02-2021", Fecha de Término: "29-04-2021", Fecha de Examen: "01-	Usuario administrador autenticado. Tener un Programa Educativo guardado previamente.	Mensaje de confirmación: "Se guardó correctamente la convocatoria"	Se despliega en la tabla el registro de la convocatoria guardada.	<input checked="" type="checkbox"/>

	07-2021", Cupo: "30"				
15	Se modifica el dato de Fecha de Inicio por : "20-02-2021" y Cupo por:"35"	Usuario administrador autenticado. Haber realizado el caso de prueba no. 14.	Mensaje de confirmación: "Se guardó correctamente la convocatoria"	Se despliega en la tabla el registro de la convocatoria modificada.	✓
16	"idConvocatoria" al hacer clic en el botón eliminar	Usuario administrador autenticado. Haber realizado el caso de prueba no. 14. Haber confirmado la eliminación de la Convocatoria en el diálogo.	Mensaje de confirmación: "Se eliminó la convocatoria"	Se actualiza la tabla sin el registro la Convocatoria eliminada.	✓
17	Requisito: "Título de licenciatura", Original/Copia:"Original", Es indispensable: "Sí", Cantidad: "1"	Usuario administrador autenticado. Haber realizado el caso de prueba no. 14. Tener por lo menos un requisito guardado previamente.	Se muestra Requisito guardado en la tabla de requisitos de la Convocatoria.	Se muestra botón para eliminar requisito si se desea.	✓

18	"idRequisitoConvocatoria" para eliminar	Usuario administrador autenticado. Haber realizado el caso de prueba no. 14. Tener por lo menos un requisito guardado previamente.	Se muestra mensaje de confirmación: "Se elimino requisito de la convocatoria"	Se actualiza la tabla sin el registro del requisito de convocatoria eliminado.	✓

Tabla 10. CP de alta, baja, y modificaciones de convocatorias.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
19	Nombre: "Título de licenciatura", Descripción: "Título que acredita la obtención del grado de licenciatura del programa educativo cursado por el aspirante", Tipo: "Documento probatorio", Documento: "Sí", Actualizable: "No"	Usuario administrador autenticado.	Se muestra mensaje de confirmación: "Se guardó requisito correctamente"	Se actualiza la tabla con el nuevo registro del Requisito ingresado.	✓
20	Se modifica el dato de Descripción por : "Documento que acredita la obtención del grado de licenciatura del programa académico cursado por el aspirante"	Usuario administrador autenticado. Haber realizado el caso de prueba no. 19	Se muestra mensaje de confirmación: "Se guardó requisito correctamente"	Se actualiza la tabla con el registro del Requisito modificado.	✓

21	"idRequisito" para eliminar	Usuario administrador autenticado. Tener por lo menos un requisito guardado previamente.	Se muestra mensaje de confirmación: "Se eliminó el requisito correctamente"	Se actualiza la tabla sin el registro eliminado.	<input checked="" type="checkbox"/>
----	-----------------------------	--	---	--	-------------------------------------

Tabla 11. CP de alta, baja y modificaciones de requisitos.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
22	Nombre: "Jorge ", Apellido: "Ramírez Pérez", Correo: "jorge@lania.edu.mx", Contraseña: "123456", clave de empleado: "JRP-1234", Rol: "Seguimiento"	Usuario administrador autenticado.	Se muestra mensaje de confirmación: "Se guardó correctamente el usuario"	Se actualiza la tabla con el nuevo registro del usuario creado.	<input checked="" type="checkbox"/>
23	Se modifica el dato de Correo por: "jorge123@lania.edu.mx"	Usuario administrador autenticado. Haber realizado el caso de prueba 22.	Se muestra mensaje de confirmación: "Se guardó correctamente el usuario"	Se actualiza la tabla con el nuevo registro del usuario modificado.	<input checked="" type="checkbox"/>
24	"idUser" eliminar para	Usuario administrador autenticado. Tener por lo menos un usuario creado.	Se muestra mensaje de confirmación: "Se eliminó correctamente el usuario"	Se actualiza la tabla con el registro del usuario eliminado.	<input checked="" type="checkbox"/>
25	Dirección de Correo: "jorge123@lania.edu.m	Haber realizado el	El usuario accede correctamente	Se muestra la pantalla de	<input checked="" type="checkbox"/>

	x", Contraseña: "123456".	caso de prueba 22.	al sistema con las credenciales proporcionadas.	dashboard de seguimiento.	
--	------------------------------	--------------------	---	---------------------------	--

Tabla 12. CP de alta, baja y modificaciones de usuarios internos.

6.1.3 Casos de prueba de funcionalidades del usuario seguimiento y administrador.

En los siguientes casos de prueba, se tomó el flujo de la validación de archivos subidos como requisitos de convocatoria en la participación del aspirante, así como la visualización del estatus de la convocatoria y estadísticas del *dashboard*.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
26	"idParticipación", "idParticipaciónReq uisitoConvocatoria", (al hacer clic en el botón validar).	Usuario seguimiento o administrador autenticado. Se tiene un aspirante participando en una convocatoria. El aspirante ha subido por lo menos un requisito de la convocatoria en su participación. Se puede visualizar el documento al hacer clic en el ícono de ojo de color verde.	Se cambia el ícono de color rojo con la etiqueta "VALIDA R" a color verde y con la etiqueta "OK"	Se cambia el conteo de requisitos validados dentro de la etiqueta de estatus de la participació n.	✓

Tabla 13. CP de usuario seguimiento valida archivos subidos por el aspirante.

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
27	"idConvocatoria", al hacer clic al	Usuario administrador	Se visualiza la información de	Se visualiza el estatus de	✓

	botón de personas de color verde.	o seguimiento autenticado. Tener por lo menos una convocatoria activa. Tener por lo menos un aspirante participando en la convocatoria.	fechas importantes de la convocatoria, el título de la convocatoria, y el listado de aspirantes participando en dicha convocatoria.	la participación de cada aspirante, en la columna "estatus".	
--	-----------------------------------	---	---	--	--

Tabla 14. CP de visualizar estatus de la convocatoria

#	Valores de entrada	Condiciones de entrada	Salida esperada	Condiciones de salida	Resultado
28		Usuario administrador o seguimiento autenticado.	Pantalla de <i>Dashboard</i> con las estadísticas de no. de aspirantes, no. de convocatorias, no. de programas educativos y listado de convocatorias a modo de tabla.	Se visualiza el no. de aspirantes registrados. Se visualiza el no. de convocatorias activas. Se visualiza el no. de programas educativos registrados. Se visualiza la tabla con el listado de convocatorias activas y botón de acción para consultar estatus de convocatoria.	✓

Tabla 15. CP de visualizar estadísticas del Dashboard.

6.2 Conclusiones del proyecto.

A continuación, se exponen los resultados obtenidos en relación a cada uno de los objetivos específicos del proyecto:

En lo relacionado al primer objetivo específico del proyecto: “**Recolectar las necesidades actuales de funcionalidad para el seguimiento al ingreso de aspirantes**”, se obtuvieron los siguientes resultados:

1. Se logró generar la documentación de las necesidades actuales de acuerdo a la Especificación de Requerimientos de Software IEEE 830-1998. (*Software Requirements Specification*) [18].
2. Se lograron detectar requerimientos actuales importantes como: visualización del estatus de la participación del aspirante, tanto del lado del aspirante como del personal de LANIA, rol de usuario de seguimiento, visualización de estadísticas por medio de un tablero, entre otros.
3. Se logró plasmar un análisis a detalle de los procesos claves de SIRA.
4. En el proceso de análisis y entrevistas con el personal del CEL, se lograron identificar ideas de mejora de interacción con los aspirantes.

Respecto al segundo objetivo específico del proyecto: “**Identificar los servicios con que debe contar la plataforma dadas las necesidades recolectadas**”, se obtuvo lo siguiente:

1. Se lograron encapsular las funcionalidades en servicios bien definidos, tomando como base los procesos clave de SIRA.
2. Como resultado del diseño de la arquitectura, se identificó la necesidad de contar con una capa de seguridad, donde se implementó *Jason Web Token* como alternativa [9].
3. Se definieron las interfaces de los servicios identificados, dotando a SIRA de características como: capacidad de interacción con otros sistemas, desacoplamiento de componentes y mayor facilidad de mantenimiento.

En relación al objetivo número tres: “**Realizar una propuesta de interfaz de usuario responsiva que brinde una adecuada visualización para móviles**” se logró lo siguiente:

1. Dotar de responsividad a la interfaz de usuario mediante componentes basados en reglas de *Material Design* [17].
2. Se logró dotar de independencia total entre el *Frontend* y *Backend* de la aplicación.
3. La facilidad de mantenimiento del cliente WEB al desacoplar en componentes su diseño, permitiendo sustituirlos o agregando más si se desea.
4. La facilidad de implementar nuevas características reactivas y de diseño responsivo al implementar el *Framework VuetifyJS* [17].

Y finalmente en cuanto al objetivo: “**Programar un prototipo que implemente los servicios y funcione con la interfaz propuesta**” se obtuvo lo siguiente:

1. Se generó un prototipo funcional que fungirá como base para un desarrollo futuro en versión de producción.
2. Se realizaron casos de prueba a los distintos escenarios funcionales del prototipo programado, obteniendo en todos ellos resultados satisfactorios.
3. Se utilizaron *frameworks*, librerías y lenguajes en sus versiones actuales, lo que permite un ciclo de vida mas amplio al momento de actualizar funcionalidades nuevas de SIRA.
4. Se implementó un control de versiones tipo GIT para facilitar el control de cambios de dicho prototipo.

6.3 Trabajos a futuro.

Como parte de trabajos a futuro del proyecto, se podrían considerar los siguientes:

- 1) Implementar alguna tecnología de chat o interacción con los aspirantes.
- 2) Cambiar el modo de autenticación por *OpenAuth* para registrarse con apps de terceros como Twitter, Facebook o Gmail.
- 3) Implementar notificaciones *push* utilizando el servicio de notificaciones.
- 4) Dotar de interacción con redes sociales para que los aspirantes puedan compartir publicaciones durante el proceso de la convocatoria.
- 5) Implementar alguna tecnología de documentación automática de los servicios de SIRA mediante herramientas como *Swagger API Blueprint*.

7 Referencias

- [1] (Olivares Rojas, 2008) Juan Carlos Olivares Rojas. *Presentación: “Arquitectura Orientada a Servicios”*. Jornadas Académicas del Instituto Tecnológico de Morelia, 2008. Recuperado el : 20-Marzo-2021 URL: [https://www.academia.edu/11789306/Arquitectura Orientada a Servicios](https://www.academia.edu/11789306/Arquitectura_Orientada_a_Servicios)
- [2] (Digital 2021) Simon Kemp. Reporte “Digital 2021: México”. Compilado por: “Kepios Pte. Ltd.”, “We are social Ltd.” y “Hootsuite Inc.”. Recuperado el 22 de marzo de 2021. Disponible en la URL: <https://datareportal.com/reports/digital-2021-mexico?rq=mexico> y en <https://www.slideshare.net/DataReportal/digital-2021-mexico-january-2021-v01> accedido: Marzo 2021.
- [3] (Vidal, P., & Martin, A. 2020). *Experiencia de Usuario + Web Responsivo: Un Estudio desde la Perspectiva de un Enfoque Integrado*. Informes Científicos Técnicos - UNPA, 12(1), 49-75. Disponible en la URL: <https://doi.org/10.22305/ict-unpa.v12.n1.703> accedido: Marzo 2021.
- [4] (BEATI, H. 2013) *Hacia una metodología de Diseño Web Responsive*, disponible en la URL: <https://es.slideshare.net/hbeati/hacia-una-metodologia-de-diseo-web-responsive>, accedido: Marzo 2021.
- [5] (CARRERA, O. 2014) *Responsive Design y accesibilidad. Buenas y malas prácticas. Errores comunes*, disponible en la URL: <https://olgacarreras.blogspot.com/2014/01/responsive-design-y-accesibilidad.html> accedido: Marzo 2021.
- [6] (O'Connor, R. V., Elger, P., Clarke, P. 2017), *Continuous Software Engineering – A Microservices Architecture Perspective*, Journal of Software:

Evolution and Process, Vol.29, No. 11, 2017. Disponible en la URL: <http://doras.dcu.ie/22111/1/ci-paper.pdf> accedido: Marzo 2021.

[7] (ZEUS, 2018), Nico Herzberg, Christoph Hochreiner, Oliver Kopp, Jörg Lenhard. 10th ZEUS Workshop, ZEUS 2018, Dresden, Germany, 8–9 February 2018 Proceedings. Challenges of Microservices Architecture: A Survey on the State of the Practice. Analyzing the Relevance of SOA Patterns for Microservice-Based Systems. Mutation Testing for Microservices. Disponible en la URL: https://www.researchgate.net/profile/Christoph-Hochreiner/publication/324517504_Proceedings_of_the_10th_ZEUS_Workshop/links/5ad1c5e9458515c60f5054d3/Proceedings-of-the-10th-ZEUS-Workshop.pdf#page=8 accedido : Marzo 2021.

[8] (Valentina Lenarduzzi and Outi Sievi-Korte. 2018). Software Components Selection in Microservices-based Systems. In Proceedings of ACM Conference (XP'18). ACM, New York, NY, USA, 3 pages. Disponible en la URL: https://www.researchgate.net/profile/Valentina-Lenarduzzi/publication/325644110_Software_Components_Selection_in_Microservices-based_Systems/links/5b1a1eddaca272021cf22be9/Software-Components-Selection-in-Microservices-based-Systems.pdf accedido : Marzo 2021.

[9] (Internet Engineering Task Force - IETF, 2015). Request For Comments (RFC) 7519. JSON Web Token (JWT). Category: Standards Track. Disponible en la URL: <https://datatracker.ietf.org/doc/html/rfc7519> accedido : Mayo 2021.

[10] (Roy Thomas Fielding, 2000). Architectural Styles and the Design of Network-based Software Architectures. Chapter 5, Representational State Transfer (REST). Disponible en la URL:

https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

accedido: Mayo 2021.

[11] (Oscar Blancarte, 2017). *Blog - Autenticación con JSON Web Tokens.*

Disponible en la URL:

<https://www.oscarblancarteblog.com/2017/06/08/autenticacion-con-json-web-tokens/> accedido: Mayo 2021.

[12] (Internet Engineering Task Force - IETF, 2005). *Request For Comments (RFC) 4051. Standards Track. Additional XML Security Uniform Resource Identifiers (URIs). Chapter 2 Algorithms. 2.1.3 SHA-384.* Disponible en la URL:

<https://www.ietf.org/rfc/rfc4051.txt> accedido : Mayo 2021.

[13] (Microsoft Corporation, 2006). *La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real.* Disponible en la URL:

https://www.academia.edu/10822644/La_Arquitectura_Orientada_a_Servicios_SO_A_de_Microsoft_aplicada_al_mundo_real?source=swp_share accedido: Abril 2021

[14] (Patricia Bazán, Laboratorio de Investigación en Nuevas Tecnologías Informáticas, 2009). *Un modelo de integrabilidad con SOA y BPM.* Disponible en la URL: <https://docplayer.es/1839609-Un-modelo-de-integrabilidad-con-soa-y-bpm.html> accedido : Abril 2021.

[15] (Comité Técnico de Arquitectura de Sistemas de Información, 2006), ARQ-RFC-01-Pautas y Recomendaciones para SOA. Disponible en la URL:

http://www.comprasesestatales.gub.uy/Aclaraciones/aclar_llamado_a157524_18.pdf accedido: Abril 2021.

- [16] (VueJs, Documentación Oficial, 2021). Sitio oficial del Framework VueJs y documentación. Disponible en la URL: <https://v3.vuejs.org/guide/a11y-resources.html#documentation> accedido: Abril 2021.
- [17] (VuetifyJs, Documentación Oficial, 2021). Sitio oficial del Framework VuetifyJs y documentación. Disponible en la URL: <https://vuetifyjs.com/en/introduction/why-vuetify/> accedido : Abril 2021.
- [18] (IEEE Computer Society, 1998), IEEE Recommended Practice for Software Requirements Specifications. Disponible en la URL : <http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf> accedido: Abril 2021.
- [19] (Marcela Castro León, Francesc Boixader, Dolores Rexachs, Emilio Luque, Universidad Autónoma de Barcelona), Arquitectura Orientada a Servicios, un enfoque basado en proyectos. Disponible en la URL: [https://www.academia.edu/28187943/Arquitectura orientada a servicios un enfoque basado en proyectos?email work card=thumbnail](https://www.academia.edu/28187943/Arquitectura_orientada_a_servicios_un_enfoque_basado_en_proyectos?email_work_card=thumbnail) accedido: Abril 2021.
- [20] (Spring Boot Framework, Documentación Oficial, 2021). Sitio oficial del Framework Spring Boot y documentación. Disponible en la URL: <https://spring.io/projects/spring-boot> accedido : Abril 2021.
- [21] (Martha Elena Zorrilla Pantaleón, Universidad de Cantabria). “Introducción a los almacenes de datos - Análisis de Drill-down”, Disponible en la URL: <https://ocw.unican.es/pluginfile.php/914/course/section/1022/Sistemas-Informacion-8.pdf> accedido: Mayo 2021.

[22] (Patricia Bazán, Universidad Nacional de La Plata). “Aplicaciones, servicios, y procesos distribuidos – Una visión para la construcción de software”, Disponible en la URL: http://sedici.unlp.edu.ar/bitstream/handle/10915/62354/Documento_completo.pdf-PDFA.pdf?sequence=1 accedido: Mayo 2021.

Anexo A. Especificaciones de casos de uso.

A1. Caso de Uso CU-01: “Crear cuenta de usuario aspirante”.

Nombre de caso de uso	CU-01 Crear cuenta de usuario aspirante			Comentarios
Prioridad	Indispensable			
Actores	Aspirante			
Eventos que inician	El aspirante desea registrarse en el sistema SIRA			
Pre-condiciones	El usuario tiene un correo electrónico válido			
Post-condiciones	El usuario recibe un correo electrónico para habilitar su cuenta			
Escenario exitoso básico	1. Aspirante ingresa a la pantalla de registro de SIRA y se muestra formulario con campos de: nombre, apellidos, correo, contraseña, no. de WhatsApp			La aplicación despliega la pantalla con el formulario de registro y botón de registrar o cancelar
	2. Aspirante ingresa su nombre completo, núm. de WhatsApp, correo electrónico, institución de procedencia, contraseña y hace clic en registrar			El sistema valida los datos ingresados y devuelve mensaje de registro exitoso
	3. Se muestra mensaje de registro exitoso			El sistema genera un token de validación y envía correo con liga para verificar la cuenta del usuario
	4. Se envía correo electrónico al usuario para validar la cuenta			
Escenario de excepción "Datos incompletos o inválidos"	2A La aplicación verifica los campos y encuentra al menos uno vacío, envía mensaje “Los dos campos son obligatorios, por favor verifique”			
	La aplicación vuelve al paso 1 y limpia los campos			
Escenario de excepción "Usuario ingresa un correo existente en SIRA"	3A La aplicación verifica el correo ingresado y ya ha sido dado de alta con anterioridad, la aplicación envía mensaje “El correo ingresado ya tiene una cuenta asociada” y muestra opción para dirigir a página de inicio de sesión			
Escenario de excepción "Link de validación excedió el tiempo"	3 A El usuario intenta verificar su cuenta 15 minutos después de generado el link			
	Mostrar opción para reenviar liga de verificación.			
Entradas	Fuentes	Salidas	Destino	
nombre, apellidos, correo, contraseña, no. WhatsApp		Mensaje de registro exitoso, pantalla de inicio de sesión		

A1. CU-01 “Crear cuenta de usuario aspirante”.

A2. Caso de uso CU-02 “ingresar al sistema”.

Nombre de caso de uso	CU-02 Ingresar al sistema			Comentarios
Prioridad	Indispensable			
Actores	Aspirante, Administrador, Seguimiento			
Eventos que inician	El aspirante, administrador o seguimiento desean ingresar al sistema			
Pre-condiciones	El usuario debe tener una cuenta válida y verificada			
Post-condiciones	El usuario accede a la vista del sistema que le corresponde			
Escenario exitoso básico	1. Aspirante, administrador o seguimiento ingresan a la pantalla de inicio de sesión de SIRA.			La aplicación despliega la pantalla con el formulario con campos de usuario, contraseña y botón de ingresar y cancelar
	2. Usuario ingresa su nombre de usuario y contraseña y da clic en ingresar			El sistema valida las credenciales y permite el acceso si éstas son correctas
	3. ¿Son correctas las credenciales?			El sistema permite o deniega el acceso al usuario
	SI. El usuario accede a su vista correspondiente	NO. El sistema envía mensaje al usuario de credenciales incorrectas		
Escenario de excepción "Datos incompletos o inválidos"	2A La aplicación verifica los campos y encuentra al menos uno vacío, envía mensaje “Los dos campos son obligatorios, por favor verifique”			
	La aplicación vuelve al paso 1 y limpia los campos			
Escenario de excepción "El usuario no recuerda contraseña"	no su	3A El usuario no recuerda su contraseña y no puede acceder		
		La aplicación muestra opción para recuperar contraseña al correo del usuario		
Entradas	Fuentes	Salidas		Destino
usuario, contraseña		Pantalla de usuario según su perfil		

A2. CU-02 “Ingresar al sistema”

A3. Caso de Uso CU-03 “CRUD de Programas Educativos”.

Nombre de caso de uso	CU-03 CRUD de Programas Educativos	Comentarios
Prioridad	Indispensable	
Actores	Administrador	
Eventos que inician	El administrador desea agregar, visualizar, modificar o eliminar Programas Educativos	
Pre-condiciones	El usuario debe tener una cuenta válida y verificada	
Post-condiciones	Se validan los datos capturados y elementos a modificar o eliminar.	
Escenario exitoso básico	1. Administrador ingresa a la pantalla de CRUD de Programas Educativos	Se muestra formulario para ingresar un nuevo registro en la parte superior, y en la parte inferior se muestra el listado con los programas Educativos existentes y botones de acción de "Modificar" y "Eliminar" para cada registro.
	2. El usuario ingresa datos del formulario: Nombre, Clave, Descripción, Vigencia.	El sistema valida los datos ingresados
	3. El usuario hace clic en el botón Crear	El sistema permite o deniega la petición si cumple o no la validación de datos. El sistema guarda los datos en la base de datos y se muestra el nuevo registro cargado en el listado de la parte inferior de la pantalla.
	4. Si el usuario hace clic en el botón Modificar en algún programa Educativo existente, vuelve al paso 1.	La aplicación muestra los datos pre cargados en el formulario del Programa Educativo a modificar.
	5. Si el usuario hace clic en el botón Eliminar algún Programa Educativo se pide confirmación.	La aplicación valida si el Programa Educativo no tienen Convocatorias asociadas para poder eliminarse. Si procede la eliminación se actualiza el listado de Programas Educativos.
Escenario de excepción "Datos incompletos o inválidos"	2A La aplicación verifica los campos obligatorios y si alguno está vacío se muestra mensaje informativo "El campo es obligatorio" en cada uno de los campos requeridos.	La aplicación alerta al usuario.
Escenario de excepción "No es posible eliminar el Programa Educativo, ya que tiene Convocatorias asociadas"	3A El programa Educativo que se desea eliminar tiene asociado Convocatorias, por lo que no es posible eliminarse y el sistema informa al usuario	La aplicación alerta al usuario.

		para que primero elimine las convocatorias asociadas a él.	
Entradas	Fuentes	Salidas	Destino
Nombre, Clave, Descripción, Vigencia.		Mensajes de operación exitosa o no exitosa.	

A3. Caso de Uso CU-03 “CRUD de Programas Educativos”.

A4. Caso de Uso CU-04 “CRUD de Convocatorias”.

Nombre de caso de uso	CU-04 CRUD de Convocatorias	Comentarios.
Prioridad	Indispensable	
Actores	Administrador	
Eventos que inician	El administrador desea agregar, visualizar, modificar o eliminar Convocatorias	
Pre-condiciones	El usuario debe tener una cuenta válida y verificada	
Post-condiciones	Se validan los datos capturados y elementos a modificar o eliminar.	
Escenario exitoso básico	1. Administrador ingresa a la pantalla de CRUD de Convocatorias	Se muestra formulario para ingresar un nuevo registro en la parte superior, y en la parte inferior se muestra el listado con las Convocatorias existentes y botones de acción de "Modificar" y "Eliminar" para cada registro. Se precarga en un combo los Programas Educativos disponibles para poder asociarlo a la convocatoria a ingresar.
	2. El usuario ingresa datos del formulario: Nombre, Descripción, Fecha de Inicio, Fecha de Término, Cantidad de Aspirantes y Programa Educativo.	El sistema valida los datos ingresados
	3. El usuario hace clic en el botón Crear	El sistema permite o deniega la petición si cumple o no la validación de datos. El sistema guarda los datos en la base de datos y se muestra el nuevo registro cargado en el listado de la parte inferior de la pantalla.
	4. Si el usuario hace clic en el botón Modificar en alguna Convocatoria existente, vuelve al paso 1.	La aplicación muestra los datos precargados en el formulario de la Convocatoria a modificar.
	5. Si el usuario hace clic en el botón Eliminar alguna Convocatoria se pide confirmación.	La aplicación valida si el Programa Educativo no tiene Aspirantes asociadas para poder eliminarse. Si procede la

		eliminación se actualiza el listado de Convocatorias.	
Escenario de excepción "Datos incompletos o inválidos"	2A La aplicación verifica los campos obligatorios y si alguno está vacío se muestra mensaje informativo "El campo es obligatorio" en cada uno de los campos requeridos.	La aplicación alerta al usuario.	
Escenario de excepción "No es posible eliminar el Programa Educativo, ya que tiene Convocatorias asociadas"	3A La Convocatoria que se desea eliminar tiene asociada Aspirantes, por lo que no es posible eliminarse y el sistema informa al usuario para que primero elimine los aspirantes asociados	La aplicación alerta al usuario.	
Entradas	Fuentes	Salidas	Destino
Nombre, Descripción, Fecha de Inicio, Fecha de Término, Cantidad de Aspirantes y Programa Educativo.		Mensajes de operación exitosa o no exitosa.	

A4. Caso de Uso CU-04 “CRUD de Convocatorias”.

A5. Caso de Uso CU-05 “Consultar convocatorias disponibles”.

Nombre de caso de uso	CU-05 Consultar convocatorias disponibles	Comentarios.	
Prioridad	Indispensable		
Actores	Aspirante		
Eventos que inician	El aspirante desea consultar las convocatorias disponibles		
Pre-condiciones	El aspirante se ha registrado y tiene una cuenta valida		
Post-condiciones			
Escenario exitoso básico	1. Aspirante ingresa a la pantalla de convocatorias, y se despliega el listado de convocatorias disponibles.	La aplicación muestra listado de convocatorias y permite navegar entre las mismas	
	2. El aspirante puede hacer clic en el botón "ver detalle" de una convocatoria para consultar los detalles, así como los requisitos necesarios.	Se muestra pantalla de detalle al hacer clic en una convocatoria	
Escenario de excepción "El servicio de convocatorias no está disponible"	2A La aplicación consulta el servicio de convocatorias, si no está disponible envía mensaje al usuario de que intente de nuevo.		
Entradas	Fuentes	Salidas	Destino

	Servicio de Convocatorias	Listado de convocatorias	Vista de convocatorias / Vista Detalle de Convocatorias
--	---------------------------	--------------------------	---

A5. Caso de Uso CU-05. "Consultar convocatorias disponibles"

A6. Caso de Uso CU-06 "Registrar Participación en Convocatoria".

Nombre de caso de uso	CU-06 Participación Convocatoria	Registrar en	Comentarios.
Prioridad	Indispensable		
Actores	Aspirante		
Eventos que inician	El aspirante puede consultar las convocatorias disponibles		
Pre-condiciones	El aspirante debe tener una cuenta válida e inicia sesión		
Post-condiciones			
Escenario exitoso básico	1. Aspirante hace clic en "Registrar Participación" dentro de la vista de detalle de convocatoria.	La aplicación muestra mensaje de registro exitoso.	
	2. El Aspirante puede acceder a la vista que permite subir la documentación de cada requisito de la convocatoria	El sistema muestra requisitos de la convocatoria, con opción de subir cada uno de los documentos y muestra el estatus de cada requisito	
	3. Se envía correo electrónico al aspirante con los detalles de la convocatoria a la que se encuentra participando.	El sistema envía correo electrónico o notificación al aspirante.	
Escenario de excepción "El servicio de registro de convocatorias no responde"	2A La aplicación intenta comunicarse al servicio de registro de participaciones de convocatorias, pero no está disponible		
	La aplicación muestra mensaje al usuario de intente de nuevo o más tarde.		
Entradas	Fuentes	Salidas	Destino
	Servicio de Participaciones	Pantalla de requisitos de convocatoria con estatus y opción para subir documentos	

A6. Caso de Uso CU-06. "Registrar Participación en Convocatoria"

A7. Caso de Uso CU-07 “Subir documentación en convocatoria”.

Nombre de caso de uso	CU-07 Subir documentación en convocatoria			Comentarios.	
Prioridad	Indispensable				
Actores	Aspirante				
Eventos que inician					
Pre-condiciones	El aspirante se encuentra registrado en una convocatoria				
Post-condiciones	El sistema debe registrar a "entregado" el estatus del requisito de la convocatoria				
Escenario exitoso básico	1. Aspirante ingresa en la vista de la convocatoria en la que está participando			La aplicación despliega la pantalla con el listado de requisitos de la convocatoria, el estatus de cada requisito y botón para subir el archivo correspondiente	
	2. El aspirante hace clic en el botón subir archivo del requisito deseado			El sistema muestra pantalla del explorador de archivos para localizar el archivo en la maquina local del usuario	
	3. El aspirante selecciona el archivo y da clic en enviar			El sistema valida que el tamaño del archivo sea menor a 2 MB y que el formato sea PDF	
	4. Aparece barra de progreso de subida del archivo				
	5. ¿El archivo se subió correctamente?				
		SI. El estatus del requisito se cambia a entregado y se visualiza en la pantalla de requisitos de la convocatoria	NO. El sistema envía mensaje al usuario de que hubo error al subir el archivo e intente de nuevo		
Escenario de excepción "No se pudo subir el archivo"	2A Existe un error de comunicación con el servidor al momento de subir el archivo				
	Se muestra mensaje informativo al usuario de que el archivo no se pudo subir, que intente nuevamente				
Escenario de excepción "El usuario no recuerda contraseña"	no su	3A El archivo que se intenta subir está en un formato no permitido			
		La aplicación muestra mensaje al usuario que el archivo debe ser en formato PDF			
Entradas	Fuentes	Salidas		Destino	
archivo PDF	Máquina local del usuario	Visualización de cambio de estatus del requisito a entregado			

A7. Caso de Uso CU-07 “Subir documentación en convocatoria”

A8. Caso de uso CU-08 “Verificar documentación del aspirante”.

Nombre de caso de uso	CU-08 Verificar documentación del aspirante		
Prioridad	Indispensable		
Actores	Seguimiento		
Eventos que inician			
Pre-condiciones	El aspirante se encuentra registrado en una convocatoria		
Post-condiciones	El usuario de seguimiento registra como completos los requisitos de la participación de la convocatoria para algún aspirante.		
Escenario exitoso básico	1. Usuario seguimiento ingresa en la vista de alguna convocatoria		La aplicación despliega la pantalla con el listado de participantes en la convocatoria
	2. El usuario seguimiento hace clic sobre alguno de los aspirantes del listado para verificar su documentación		El sistema despliega pantalla de requisitos entregados por el aspirante en la convocatoria
	3. Por cada documento requisito ingresado por el aspirante		
	El usuario seguimiento hace clic en uno de los documentos requisito para verificar que sea correcto		
	¿Es correcto el documento o requisito?		
	SI. El usuario seguimiento da clic en el botón para asignar estatus de completado	NO. Se deja el estatus sin completar	El sistema guarda el estatus de completado en el caso que sea marcado por el usuario de seguimiento
	4. ¿Se han validado todos los documentos del aspirante?		
	SI. Se envía correo de notificación de estatus completado al aspirante.	No.	si el estatus de la participación es completado, se envía un correo o notificación al aspirante .
Escenario de excepción "No se pudo consultar la lista de convocatorias"	1A Existe un error de comunicación con el servidor al momento de consultar el Servicio de Participaciones		
	Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones, que intente nuevamente		
Escenario de excepción "No se pudo consultar la	2A Existe un error de comunicación con el servidor al momento de consultar el servicio de Participaciones		

lista de participantes o de documentos requisito"		Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones, que intente nuevamente	
Entradas	Fuentes	Salidas	Destino
Servicio de Participaciones		Visualización de cambio de estatus del requisito a entregado	

A8. Caso de Uso CU-08 “Verificar documentación del aspirante”

A9. Caso de uso CU-09 “Consultar estado de convocatorias del aspirante”.

Nombre de caso de uso	CU-09 Consultar estado de convocatorias del aspirante	Comentarios.
Prioridad	Indispensable	
Actores	Seguimiento, Aspirante	
Eventos que inician		
Pre-condiciones	El aspirante se encuentra registrado en una convocatoria	
Post-condiciones		
Escenario exitoso por parte del usuario Seguimiento	1. Usuario seguimiento ingresa en la vista de alguna convocatoria 2. El usuario seguimiento puede visualizar el estatus de cada participación (por aspirante)	La aplicación despliega la pantalla con el listado de participantes en la convocatoria
Escenario exitoso por parte del usuario Aspirante	1A. El aspirante ingresa a la vista "Mis Participaciones"	El sistema despliega el listado de las convocatorias donde se encuentra participando el aspirante.
	2A. El aspirante puede visualizar de manera rápida en cada participación el estatus de las mismas	
	3.A El aspirante hace clic en alguna participación para ver el detalle de sus requisitos	La aplicación despliega el detalle del estatus de los requisitos de su participación
Escenario de excepción "No se pudo consultar la lista de convocatorias"	1A Existe un error de comunicación con el servidor al momento de consultar el Servicio de Participaciones	
	Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones, que intente nuevamente	
Escenario de excepción "No se pudo consultar la lista de	2A Existe un error de comunicación con el servidor al	

participantes o de documentos requisito"		momento de consultar el servicio de Participaciones	
		Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones, que intente nuevamente	
Entradas	Fuentes	Salidas	Destino
	Servicio de Participaciones	Visualización estatus de las participaciones y de los requisitos en las vistas de detalle	

A9. Caso de Uso CU-09 “Consultar estado de convocatorias del aspirante”.

A10. Caso de uso CU-10 “Consultar estatus general de convocatorias”.

Nombre de caso de uso	CU-10 Consultar estatus general de las convocatorias		
Prioridad	Indispensable		
Actores	Administrador, Seguimiento		
Eventos que inician			
Pre-condiciones	Existen convocatorias activas o vigentes, el usuario se autentifica correctamente		
Post-condiciones			
Escenario exitoso por parte del usuario Seguimiento	1. Usuario seguimiento o administrador ingresa al sistema con usuario y contraseña validos o bien hace clic en la opción "Dashboard"		La aplicación despliega la pantalla con el tablero o Dashboard con las estadísticas de convocatorias activas
	2. El usuario seguimiento o administrador visualiza el no. de convocatorias activas, no de aspirantes y estatus general de convocatorias.		La aplicación despliega estadísticas sobre el estatus de las convocatorias activas
Escenario de excepción "No se pudo consultar la lista de convocatorias"	1A Existe un error de comunicación con el servidor al momento de consultar el Servicio de Participaciones		
	Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones , que intente nuevamente		
Entradas	Fuentes	Salidas	Destino
	Servicio de Participaciones		

A10. Caso de Uso CU-10 “Consultar estatus general de convocatorias”.

A11. Caso de uso CU-11 “consumir datos generales de aspirantes”.

Nombre de caso de uso	CU-13 Consumir datos generales de aspirantes	Comentarios	
Prioridad	Indispensable		
Actores	Sistema SICEL o cualquier sistema externo		
Eventos que inician			
Pre-condiciones	Existen datos de aspirantes en la base de datos de SIRA, el cliente tiene un token JWT válido		
Post-condiciones	Se obtengan datos generales de los aspirantes en formato JSON		
Escenario exitoso por parte del usuario Seguimiento	1. Sistema SICEL o cualquier cliente externo genera una petición de tipo GET por medio del protocolo HTTP a la URI "/aspirantes" 2. Se genera un objeto respuesta en formato JSON con los datos generales de cada uno de los aspirantes	En el encabezado de la petición deberá utilizar autenticación con JWT y el token válido La aplicación regresa el objeto JSON con la información	
Escenario de excepción "No se pudo consultar la lista de convocatorias"	1A Existe un error de comunicación con el servidor al momento de consultar el Servicio de Participaciones Se muestra mensaje informativo al usuario de que no se pudo comunicar al servicio de participaciones , que intente nuevamente		
Entradas	Fuentes	Salidas	Destino
token JWT	Servicio de Usuarios	Objeto en formato JSON con los datos generales de los aspirantes	

A11. Caso de Uso CU-11 “Consumir datos generales de aspirantes”.

A12. Caso de uso CU-12 “CRUD requisitos”.

Nombre de caso de uso	CU-14 CRUD Requisitos	Comentarios
Prioridad	Indispensable	
Actores	Administrador	
Eventos que inician	El administrador desea agregar, visualizar, modificar o eliminar Requisitos	
Pre-condiciones	El usuario debe tener una cuenta válida y verificada	
Post-condiciones	Se validan los datos capturados y elementos a modificar o eliminar.	
Escenario exitoso básico	1. Administrador ingresa a la pantalla de CRUD de Requisitos	Se muestra formulario para ingresar un nuevo registro en la parte superior, y en la parte inferior se muestra el listado

		con los Requisitos existentes y botones de acción de "Modificar" y "Eliminar" para cada registro.	
	2. El usuario ingresa datos del formulario: Nombre, Descripción, tipo y si es documento o no.	El sistema valida los datos ingresados	
	3. El usuario hace clic en el botón Guardar	El sistema permite o deniega la petición si cumple o no la validación de datos. El sistema guarda los datos en la base de datos y se muestra el nuevo registro cargado en el listado de la parte inferior de la pantalla.	
	4. Si el usuario hace clic en el botón Modificar en algún requisito existente, vuelve al paso 1.	La aplicación muestra los datos pre cargados en el formulario del requisito a modificar.	
	5. Si el usuario hace clic en el botón Eliminar algún requisito se pide confirmación.	La aplicación valida si el requisito no tiene Convocatorias asociadas para poder eliminarse. Si procede la eliminación se actualiza el listado de requisitos.	
Escenario de excepción "Datos incompletos o inválidos"	2A La aplicación verifica los campos obligatorios y si alguno está vacío se muestra mensaje informativo "El campo es obligatorio" en cada uno de los campos requeridos.	La aplicación alerta al usuario.	
Escenario de excepción "No es posible eliminar el Requisito, ya que tiene Convocatorias asociadas"	3A El requisito que se desea eliminar tiene asociada Convocatorias, por lo que no es posible eliminarse y el sistema informa al usuario para que primero elimine su asociación de la convocatoria	La aplicación alerta al usuario.	
Entradas	Fuentes	Salidas	Destino
Nombre, Descripción, tipo y si es documento o no.		Mensajes de operación exitosa o no exitosa.	

A12. Caso de Uso CU-12 “CRUD Requisitos”

A13. Caso de uso CU-13 “Asignar requisitos a convocatorias”

Nombre de caso de uso	CU-15 Asignar requisitos a convocatorias	Comentarios
Prioridad	Indispensable	

Actores	Administrador		
Eventos que inician	El administrador desea agregar o quitar requisitos a convocatorias		
Pre-condiciones	El usuario debe tener una cuenta válida y verificada, debe haber requisitos guardados en la base de datos		
Post-condiciones			
Escenario exitoso básico	<p>1. Administrador ingresa a la pantalla de CRUD de Convocatorias</p> <p>2. El administrador hace clic en el icono de acción de agregar requisitos a convocatoria</p> <p>3. El administrador selecciona del combo de requisitos disponibles alguno para asociar a la convocatoria, además de ingresar datos de: cantidad, original, copia o ambos, y si es indispensable o no.</p> <p>4. El administrador hace clic en el botón Agregar requisito.</p> <p>5. Si el usuario hace clic en el botón Quitar requisito en alguno de los requisitos listados en la parte superior del diálogo, aparecerá un mensaje de confirmación para desasociar dicho requisito de la convocatoria</p>	<p>Se muestra la tabla con el listado de Convocatorias actuales, en la columna acciones se despliegan diversos iconos de acción entre ellos el de asignar requisito a convocatoria</p> <p>El sistema despliega diálogo con el listado de requisitos asociados a la convocatoria y formulario para agregar más</p> <p>El sistema carga en un combo los requisitos disponibles guardados previamente</p> <p>El sistema guarda la asociación y actualiza el listado de requisitos asociados.</p> <p>Si el usuario confirma, se desasociará el requisito en cuestión de la convocatoria .</p>	
Escenario de excepción "Datos incompletos o inválidos"	<p>2A La aplicación verifica los campos obligatorios y si alguno está vacío se muestra mensaje informativo "El campo es obligatorio" en cada uno de los campos requeridos.</p>	La aplicación alerta al usuario.	
Escenario de excepción "No es posible eliminar el Requisito, hubo un error al comunicarse al servicio de catálogos"	<p>3A La comunicación al servicio de catálogos no está disponible, por lo que se alerta al usuario que intente de nuevo o más tarde.</p>	La aplicación alerta al usuario.	
Entradas	Fuentes	Salidas	Destino
el requisito, cantidad, si es original, copia o ambos, si es indispensable o no.	Servicio de Catálogos	Mensajes de operación exitosa o no exitosa.	

A13. Caso de Uso CU-13 “Asignar requisitos a convocatorias”

A14. Caso de uso CU-14 “CRUD de Usuarios”.

Nombre de caso de uso	CU-14 CRUD Usuarios		
Prioridad	Indispensable		
Actores	Administrador		
Eventos que inician	El administrador desea agregar, visualizar, modificar o eliminar Usuarios internos		
Pre-condiciones	El usuario debe tener una cuenta válida y verificada		
Post-condiciones	Se validan los datos capturados y elementos a modificar o eliminar.		
Escenario exitoso básico	1. Administrador ingresa a la pantalla de CRUD de Usuarios		Se muestra formulario para ingresar un nuevo usuario en la parte superior, y en la parte inferior se muestra el listado con los Requisitos existentes y botones de acción de "Modificar" y "Eliminar" para cada registro.
	2. El usuario ingresa datos del formulario: Nombre, apellido, correo, contraseña, clave de empleado y rol.		El sistema valida los datos ingresados
	3. El usuario hace clic en el botón Guardar		El sistema permite o deniega la petición si cumple o no la validación de datos. El sistema guarda los datos en la base de datos y se muestra el nuevo registro cargado en el listado de la parte inferior de la pantalla.
	4. Si el usuario hace clic en el botón Modificar en algún usuario existente, vuelve al paso 1.		La aplicación muestra los datos pre cargados en el formulario del usuario a modificar.
	5. Si el usuario hace clic en el botón Eliminar algún usuario se pide confirmación.		Si procede la eliminación se actualiza el listado de usuarios.
Escenario de excepción "Datos incompletos o inválidos"		2A La aplicación verifica los campos obligatorios y si alguno está vacío se muestra mensaje informativo "El campo es obligatorio" en cada uno de los campos requeridos.	
		La aplicación alerta al usuario.	
Entradas	Fuentes	Salidas	Destino
Nombre, apellido, correo, contraseña, clave de empleado, rol.		Mensajes de operación exitosa o no exitosa.	

A14. Caso de Uso CU-14 “CRUD de Usuarios”