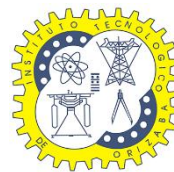




EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



**Tecnológico Nacional de México
Instituto Tecnológico de Orizaba**

**División de Estudios de Posgrado e Investigación
Maestría en Sistemas Computacionales**

TESIS

TÍTULO DEL PROYECTO:

Plataforma colaborativa para dar seguimiento a la metodología Prodigios del
Territorio.

PARA OBTENER EL GRADO DE:

Maestro en Sistemas Computacionales

PRESENTA:

I.S.C. Adolfo Meza Romero

DIRECTOR DE TESIS:

M.C.E. Beatriz Alejandra Olivares Zepahua

CODIRECTOR DE TESIS:

Dra. Elsa Antonia Pérez Paredes

Febrero 2024



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, **20/febrero/2024**
Dependencia: **División de Estudios de
Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

C. ADOLFO MEZA ROMERO
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES
P R E S E N T E.-

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" Plataforma colaborativa para dar seguimiento a la metodología prodigios del territorio"

comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

DRA. OFELIA LANDETA ESCAMILLA
ENCARGADA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN

OG-13-F06



Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.
Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx



2024
Felipe Carrillo
PUERTO
PRESIDENTE DEL PROLETARIADO
REVOLUCIONARIO Y DEFENSOR
DEL MEXICO



Orizaba, Veracruz, 09/febrero/2024
Asunto: Revisión de trabajo escrito

C. OFELIA LANDETA ESCAMILLA
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
P R E S E N T E.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

ADOLFO MEZA ROMERO

La cual lleva el título de:

Plataforma colaborativa para dar seguimiento a la metodología prodigios del territorio

Y concluyen que se acepta.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

PRESIDENTE: M.C.E. BEATRIZ ALEJANDRA
OLIVARES ZEPAHUA


FIRMA

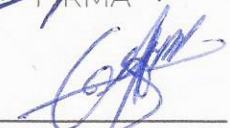
SECRETARIO: DR. JOSÉ LUIS SÁNCHEZ
CERVANTES


FIRMA

VOCAL: M.S.C. LUIS ÁNGEL REYES
HERNÁNDEZ


FIRMA

VOCAL SUP.: DRA. ELSA ANTONIA PÉREZ
PAREDES


FIRMA

TA-09-18



Declaración de originalidad y cesión de derechos

Orizaba, Veracruz, el día 24 del mes de Noviembre del año 2023.

El(la) que suscribe

C. Adolfo Meza Romero

Declaro que esta tesis, que tiene una extensión de 103 cuartillas, ha sido escrita por mí y constituye el registro escrito del trabajo de la tesis titulada

"Plataforma colaborativa para dar seguimiento a la metodología Prodigios del Territorio"

del programa: Maestría en Sistemas Computacionales bajo la asesoría y dirección de la M.C.E. Beatriz Alejandra Olivares Zepahua y la Dra. Elsa Antonia Pérez Paredes y no ha sido sometida en ninguna otra institución previamente.

Todos los datos y las referencias a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal y, en los casos que así lo requieran, cuento con las debidas autorizaciones de quienes poseen los derechos patrimoniales. Por lo tanto, me hago responsable de cualquier litigio o reclamación relacionada con derechos de propiedad intelectual, exonerando de toda responsabilidad al Tecnológico Nacional de México campus Orizaba.

También declaro que, al presentar esta tesis, cedo los derechos del trabajo al Tecnológico Nacional de México campus Orizaba para su difusión, con fines académicos y de investigación, bajo las regulaciones propias de la institución y que si existe algún acuerdo de confidencialidad de la información lo haré saber en forma escrita para que se omitan las secciones correspondientes.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: depi_orizaba@tecnm.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente de este.



Adolfo Meza Romero

Nombre y Firma

Agradecimientos

Doy mis agradecimientos a Dios por todas las bendiciones y oportunidades que se me han otorgado, así como a las personas que puso en mi camino pues es gracias a ellas que me encuentro en este punto de mi vida.

Agradezco a mis padres, pues gracias a su sacrificio y constante apoyo he podido llevar una vida acomodada aun dentro de sus posibilidades. Les agradezco por enseñarme el sentido de la responsabilidad, así como los valores que han forjado las bases de la persona que soy el día de hoy. Realmente no puedo expresar en palabras todo lo que han hecho por mi y siempre estaré en deuda con ustedes. Este logro va dedicado a ustedes y a mis abuelos, aunque ya no se encuentren entre nosotros.

Doy gracias a mis amigos, pues ellos han significado un gran apoyo moral y una influencia positiva en mi vida. Especialmente, agradezco a Ana, Raúl y Pedro por formar parte de mi vida durante mi paso por el Tecnológico de Orizaba y estar conmigo durante algunos de los mis mejores años.

Agradezco a CONAHCYT por el apoyo económico proporcionado durante la realización de esta tesis, pues dicho apoyo resultó fundamental para su finalización.

Agradezco al Tecnológico Nacional de México, campus Orizaba, por ofrecer a los mejores docentes, quienes fungieron como guía durante mi educación.

Agradezco al equipo que formó parte del desarrollo de este proyecto, conformado por los alumnos de residencias profesionales Claudia, Jocelyn y Braulio, y la estudiante de servicio social Angélica.

Agradezco a la doctora Elsa Antonia Pérez Paredes por su interés y ayuda en el encaminamiento del tema de tesis, pues su perspectiva y experiencia derivada de áreas más orientadas a las humanidades no habría sido posible fundamentar las bases teóricas de la presente tesis.

Agradezco, también, a la maestra Beatriz Alejandra Olivares Zepahua por su constante apoyo y paciencia a lo largo de la realización de esta tesis, pues sin su orientación y disposición para resolver los problemas surgidos durante este trabajo no habría sido posible su culminación.

Índice General	
Índice de Figuras	iii
Índice de Tablas	iv
Índice de Listados	iv
Resumen	v
Abstract	vi
Introducción	1
Capítulo 1 Antecedentes	2
1.1 Marco teórico	2
1.1.1 Naturaleza	2
1.1.2 Territorio	2
1.1.3 Biocultural	3
1.1.4 Ambiente	3
1.1.5 Educación Formal	4
1.1.6 Educación no formal	4
1.1.7 Educación ambiental crítica	5
1.1.8 Prodigio del Territorio	5
1.1.9 Metodología de Prodigios del Territorio	6
1.1.10 Storytelling	7
1.1.11 Escritura funcional y creativa	8
1.1.12 Participación ciudadana	8
1.1.13 Apropiación del Territorio	9
1.1.14 Aplicación Web	9
1.1.15 Plataforma Colaborativa	10
1.1.16 Aplicación de una sola página	10
1.1.17 Arquitectura de múltiples capas	11
1.2 Planteamiento del problema	12
1.3 Objetivo general y específicos	14
1.4.1 Objetivo General	14
1.4.2 Objetivos Específicos	14
1.4 Justificación	14
Capítulo 2 Estado de la práctica	16
2.1 Trabajos relacionados	16

2.2	Análisis comparativo.....	25
2.3	Propuesta de Solución.....	28
2.3.1	Angular	34
2.3.2	Django	35
2.3.3	PostgreSQL	35
2.3.4	Digital Ocean	36
2.3.5	Adaptación de Scrum	36
Capítulo 3	Aplicación de la metodología.....	38
3.1	Descripción de la solución.....	38
3.2	Metodología de desarrollo Scrum	38
3.2.1	Sprint 1. Análisis.....	38
3.2.2	Sprint 2. Diseño de la Arquitectura	45
3.2.3	Sprint 3. Diseño de la aplicación	47
3.2.4	Sprint 4. Codificación de los servicios de Django Rest Framework	58
3.2.5	Sprint 5. Codificación de la aplicación Angular	69
3.2.6	Sprint 6. Pruebas de la ejecución del proceso de la plataforma.....	84
Capítulo 4	Resultados	87
4.1	Caso de Estudio: Encuentro Comunitario de Prodigios del Territorio	87
4.1.1	Planteamiento del caso de estudio	87
4.1.2	Características del caso de estudio	87
4.1.3	Descripción de las actividades realizadas.....	88
4.1.4	Resultados obtenidos del caso de estudio	91
Capítulo 5	Conclusiones y Recomendaciones	93
5.1	Conclusiones.....	93
5.2	Recomendaciones.....	94
Anexos y Apéndices.....		96
Anexos.....		96
Anexo A: Carta de Usuario		97
Referencias		98

Índice de Figuras

Figura 1.1 Ejemplo de arquitectura de múltiples capas	11
Figura 2.1 Arquitectura preliminar de la solución.....	28
Figura 2.2 Bosquejo del funcionamiento de la plataforma (Parte 1)	30
Figura 2.3 Bosquejo del funcionamiento de la plataforma (Parte 2)	31
Figura 2.4 Bosquejo del funcionamiento de la plataforma (Parte 3)	32
Figura 3.1 Diagrama de casos de uso general	39
Figura 3.2 Sub diagrama de casos de uso, Gestionar Consejo	41
Figura 3.3 Sub diagrama de Casos de Uso, Gestionar Convocatorias.....	42
Figura 3.4 Diagrama de clases.....	42
Figura 3.5 Arquitectura Final del Sistema (Diagrama de Componentes)	47
Figura 3.6 Pantalla principal de la plataforma.....	48
Figura 3.7 Pantalla principal codificada en Angular	48
Figura 3.8 Diagrama entidad relación de la base de datos	54
Figura 3.9 Estructura de los archivos del proyecto (servidor)	60
Figura 3.10 Contenido de la carpeta "modelos"	61
Figura 3.11 Contenido del archivo __init__.py	61
Figura 3.12 Modelo Actualizado de la Base de Datos.....	63
Figura 3.13 Ejecución del archivo quickstart.py	68
Figura 3.14 Pantalla de autenticación para generar credenciales del API de Google Drive....	68
Figura 3.15 Estructura de la carpeta platform.	70
Figura 3.16 Estructura del módulo de Usuario Participante.	71
Figura 3.17 Estructura del Componente Registrar Prodigio.	71
Figura 3.18 Contenido del archivo constantes	73
Figura 3.19 Carrusel de imágenes de la galería.....	85
Figura 3.20 Listado de municipios (previo a la corrección)	86
Figura 4.1 Presentación sobre el planteamiento de la plataforma.	88
Figura 4.2 Presentación sobre el proceso de la plataforma.	88
Figura 4.3 Explicación sobre la edición de imágenes.....	89
Figura 4.4 Fotografía de parte del equipo de asistencia tecnológica.	90

Figura 4.5 Fotografías del equipo brindando orientación a los participantes.	90
Figura 4.6 Presentación de la plataforma en línea.	91

Índice de Tablas

Tabla 2.1 Análisis comparativo de los artículos revisados	25
Tabla 3.1 Comparación de tamaños de archivo y tiempos de conversión	46
Tabla 3.2 Diseño de los servicios del Caso de uso “Ver Galería”	50

Índice de Listados

Listado 3.1 Contenido interno de una clase del proyecto.....	62
Listado 3.2 Código de una de las vistas de la aplicación	64
Listado 3.3 Ejemplo de inserción empleando los métodos heredados del modelo	65
Listado 3.4 Incorporación de proceso almacenados en la aplicación	66
Listado 3.5 Contenido del archivo de settings.yaml.	67
Listado 3.6 Código para la generación del módulo de credenciales.....	67
Listado 3.7 Función para la subida de archivos de vídeo.....	69
Listado 3.8 Importación y Uso de la clase HttpClient.	72
Listado 3.9 Distintas formas del uso de la función get.	74
Listado 3.10 Ejemplo del uso de la función post.	75
Listado 3.11 Ejemplo del uso de la función patch.....	76
Listado 3.12 Ejemplo de Ruta definida en el módulo de enrutamiento principal.	77
Listado 3.13 Ejemplo de Ruta definida en el módulo de participante.	77
Listado 3.14 Contenido del módulo de enrutamiento del componente Corregir Prodigio.	78
Listado 3.15 Módulo de enrutamiento del componente Subir Prodigio.	79
Listado 3.16 Importaciones y su implementación en el componente Corregir Prodigio.	80
Listado 3.17 Ejemplo del uso de servicios en la función listarCategorias.	81
Listado 3.18 Presentación de multimedia en la plataforma.....	82
Listado 3.19 Pipe codificada para la sanitización de enlaces embebidos.	83
Listado 3.20 Presentación de videos en la plataforma.	84

Resumen

En la actualidad, las problemáticas ambientales y el agotamiento de las instituciones ante las mismas generaron la necesidad de fomentar el interés y la integración de la sociedad civil en las actividades de preservación del territorio, esto a través de una educación ambiental que funcione como un proceso de intercambio de conocimientos, experiencias y simbolismos del territorio que considere al individuo como un actor capaz de generar un cambio positivo en su realidad ambiental.

Pese a que en México existen estrategias colegiadas que atienden esta necesidad, no suelen alcanzar la visibilidad deseada, pues se quedan en círculos específicos y pasan desapercibidas para la población general. Para abordar esta situación, se propuso el desarrollo de una plataforma colaborativa que, siguiendo las pautas de la Metodología de “Prodigios del Territorio”, funcione como un espacio para la recolección y difusión de los contenidos afines a este paradigma de educación integral.

La plataforma propuesta se desarrolló como una aplicación web, por medio de la cual diversos convocantes tendrán la capacidad de abrir espacios para la recepción de narrativas de prodigios del territorio, que se subirán en formato de audio, vídeo, imagen o documento. Una vez cerrada la recepción de prodigios, los organizadores de las convocatorias realizarán un proceso de selección y curado de los contenidos recibidos para su posterior exposición dentro de la plataforma.

Abstract

Currently, environmental problems and the exhaustion of institutions in the face of them generated the need to promote the interest and integration of civil society in territorial preservation activities, through an environmental education that works as a process that involves the exchange of knowledge, experiences and symbolism of the territory, considering the individual as an actor capable of generating a positive change in their environmental reality.

Although in Mexico there are collegiate strategies that address this need, they do not achieve the desired visibility, since they remain in specific circles and go unnoticed by the general population. To address this situation, the development of a collaborative platform was proposed. Following the guidelines of the “Prodigies of the Territory” Methodology, the platform works as a space for the collection and broadcasting of content related to this paradigm of comprehensive education.

The proposed platform was developed as a web application, through which various conveners will have the opportunity to open spaces for the reception of narratives of wonders of the territory, which will be uploaded in audio, video, image or document format. Once the reception of prodigies is closed, the organizers of the calls will carry out a selection and curation process of the content received for subsequent exhibition within the platform.

Introducción

Debido a las problemáticas ambientales enfrentadas en la actualidad, se volvió necesario el fomentar una educación ambiental que promueva la participación de la ciudadanía en actividades de reconocimiento, resignificación y conservación del territorio. Con este fin, se requiere que la educación funcione como un proceso social en donde se intercambien conocimientos, experiencias y simbolismo, de modo que toda la información disponible sea aprovechada para generar un impacto positivo en la realidad ambiental. Actualmente existen estrategias colegiadas que responden a las problemáticas ambientales empleando este enfoque social hacia el territorio, sin embargo, en la mayoría de las ocasiones pasan desapercibidas para la población.

Con base en lo anterior, el presente proyecto de tesis busca generar un espacio digital para la recolección y difusión de contenidos afines a este enfoque educativo integral y de participación ciudadana. Para este propósito, se planteó el desarrollo de una plataforma colaborativa que dé seguimiento a la metodología de “Prodigios del Territorio”, una estrategia de reconocimiento y apropiación del territorio, a través del uso de tecnologías web.

El presente documento se estructura de la siguiente forma: En el capítulo uno, Antecedentes, se exponen los conceptos más relevantes para la definición del proyecto, así como su objetivo general, objetivos específicos y justificación. Dentro del segundo capítulo, Estado de la Práctica, se resumen los trabajos que presentan un contenido similar al del presente proyecto, enfocándose en la problemática tratada, la solución propuesta y los resultados obtenidos, realizando una comparación de los trabajos con una mayor similitud al tema tratado en este proyecto y, al final, se describe la propuesta de solución. El tercer capítulo, Aplicación de la Metodología, describe la implementación de la solución propuesta en capítulo dos, siguiendo la metodología seleccionada. En el capítulo cuatro, Resultados, se expone el caso de estudio realizado para probar la usabilidad del producto de software generado. Finalmente, en el capítulo cinco, Conclusiones y Recomendaciones, se exponen las consideraciones finales derivadas del desarrollo implementado y los resultados obtenidos.

Capítulo 1 Antecedentes

En este capítulo se presenta el marco teórico, en el cual se definen los conceptos más relevantes para la contextualización del proyecto de tesis y, además, se establece el planteamiento del problema, el objetivo general, los objetivos específicos y la justificación.

1.1 Marco teórico

A continuación, se explican los términos más relevantes relacionados con el proyecto de tesis.

1.1.1 Naturaleza

El concepto de “naturaleza” es un término que, a lo largo de su historia, agrega significados distintos que en ocasiones generan conflictos unos con otros. Pese a esto, en la mayoría de los conceptos, tanto orientales como occidentales, se encuentra presente la idea de que la naturaleza es ajena a la influencia y acción humana. En las concepciones académicas actuales, la naturaleza tiene cuatro definiciones dentro de los diccionarios [1]:

- El conjunto de la realidad material, considerado como independiente de la actividad humana y de la historia.
- Todo el universo, ya que es el lugar, la fuente y el resultado de fenómenos materiales (incluyendo hombre o al menos el cuerpo del hombre).
- La fuerza específica en el centro de la vida y el cambio.
- La esencia, cualidad interior y carácter, el todo de propiedades físicas específicas de un objeto, vivo o inerte.

Para este proyecto, tomando en cuenta las definiciones anteriores, la naturaleza es todo lo que compone el mundo o universo material, incluyendo todo fenómeno, materia inerte y vida, originados por los procesos intrínsecos del mundo, ajenos a la intervención humana, que pertenecen a dicho universo.

1.1.2 Territorio

El término “territorio” suele tener una noción confusa debido a su polisemia, originada por las distintas perspectivas que se tienen del término provenientes de diversas disciplinas como, por ejemplo, la biología, la ecología, la arquitectura y la economía. En este proyecto de tesis, el territorio toma en consideración varios elementos; por ejemplo, las plantas ocupan y delimitan

ciertas zonas con base en las características del suelo y la disponibilidad del agua, los animales construyen nidos, establecen zonas de caza y, empleando sus sonidos u olores corporales, delimitan su territorio [2] y, por parte del ser humano, el avance cultural y tecnológico, la apropiación de recursos y los derechos de propiedad, dejan una huella en las configuraciones espaciales que alojan las actividades del ser humano, lo que permite establecer un límite para lo que la población reconoce como territorio.

Teniendo en cuenta lo expuesto anteriormente, el territorio se define como la representación del espacio que: posee características naturales y configuraciones espaciales específicas, contiene las prácticas sociales y los sentidos simbólicos que los seres humanos desarrollan en sociedad en su íntima relación con la naturaleza [3], y que se ve sometido a una transformación continua que resulta de la acción social de los seres humanos, su cultura y las revoluciones del conocimiento que se viven en todo el mundo.

1.1.3 Biocultural

El concepto “biocultural” posee distintas definiciones dependiendo de la disciplina desde la que se aborda. Por ejemplo, se denominan regiones bioculturales a los territorios de pueblos indígenas que coinciden con centros de origen y diversificación de biodiversidad, así como de diversidad doméstica; por otro lado, los recursos bioculturales son especies de plantas, animales y hongos que forman parte del uso, manejo y conocimiento tradicional de las comunidades [4]. Pese a esta diversidad de definiciones, existe un consenso general que establece el término “biocultural” como la asociación entre las diversidades biológica, lingüística y cultural.

Considerando lo anterior, “biocultural” establece de forma directa una relación entre la diversidad biológica y la diversidad cultural, que se encuentra influenciada por las configuraciones espaciales y temporales de un determinado lugar y está sujeta a la herencia y tradiciones de una cultura en particular.

1.1.4 Ambiente

Dada la extensa variedad de perspectivas y disciplinas que tratan con el término ambiente y la complejidad de la noción del término, es necesario establecer una perspectiva interdisciplinaria que permita fomentar valores éticos, económicos y estéticos junto con la participación

individual y comunitaria, y enfatizar la relación existente entre sociedad y naturaleza. Con esta perspectiva en mente, el ambiente se define como un sistema complejo en donde interactúan los procesos biofísicos propios de la naturaleza y las relaciones sociales en forma bidireccional, los cuales se encuentran mediados por la cultura [5].

1.1.5 Educación Formal

Se entiende como educación formal a la modalidad educativa impartida en las instituciones que conforman un sistema escolar oficial autorizado (público o privado). Comúnmente, se transmite desde establecimientos propios, coordinados por autoridades gubernamentales. Debido a esto, la administración educativa se organiza como una jerarquía, contando con un marco legal específico para cada país. Por su parte, el personal docente es principalmente profesional o, al menos, reconocido oficialmente como tal.

En esta modalidad, las instituciones se organizan por grados, ciclos, niveles y modalidades, lo que formaliza una estructura y especialización escalonada. El progreso del estudiante se mide a través un sistema de créditos, grados, títulos y certificados que avalan los logros académicos obtenidos, y conforman parte de los requisitos de acceso al siguiente nivel. Los planes educativos son generales y enfocados a una enseñanza integral, además, sus metas se planifican con base en el desarrollo intelectual y la formación física, social y afectiva del estudiante [6].

1.1.6 Educación no formal

La educación no formal se trata de un concepto relativamente reciente, consecuencia del mejoramiento de las condiciones de vida de las sociedades occidentales más desarrolladas. En función de precisar el ámbito de acción de la educación no formal, es importante conocer e identificar las diferencias entre esta modalidad educativa y las modalidades formal e informal. La educación formal ya se trató en el punto anterior, por otro lado, la educación informal se refiere a un proceso que ocurre durante toda la vida, en el que las personas adquieren conocimientos, habilidades, actitudes y modos de discernimiento por medio de experiencias diarias.

La educación no formal involucra actividades organizadas, sistemáticas y educativas realizadas fuera del marco del sistema educativo oficial, es decir, se define como un proceso educativo

planificado, que no necesariamente conlleva la obtención de un título o grado académico, con una duración y objetivos definidos, estructurado en sus acciones, contenidos y métodos, y desarrollado por instituciones y organizaciones de ámbitos diversos [7]. Además, la educación no formal:

- Surge de una problemática social, siendo la propia sociedad quien propone y lleva a cabo soluciones.
- Se dirige a todos los sectores de la población, adaptándose a los procesos de aprendizaje propios de cada caso.
- Posee una planificación y sistematización mínima en comparación con la educación formal.
- Es capaz de fomentar una conciencia ante situaciones, problemas o necesidades emergentes.
- Asume fácilmente nuevas innovaciones metodológicas, lo que permite propuestas formativas dinámicas, activas y atractivas.

1.1.7 Educación ambiental crítica

La educación ambiental, como práctica social crítica, surge a partir de la necesidad de considerar al individuo como un actor social que, mediante la visión y entendiendo la complejidad de los problemas ambientales, logre la transformación social de su entorno para generar una realidad ambiental distinta. Por ende, se requiere de una educación que permita definir una postura que logre una transformación social, según el contexto ambiental particular, y fomente la participación de la sociedad civil en la generación de alternativas a los problemas ambientales locales, contemplando los procesos históricos, sociales y económicos que provocaron la degeneración ambiental [5].

1.1.8 Prodigio del Territorio

El término prodigio hace referencia a un evento anormal que sobrepasa los límites establecidos de la naturaleza, un milagro. Tomando como base esta definición, un prodigio del territorio se entiende como un hecho que existe pese a las limitaciones regulares de una determinada zona y las adversidades de su entorno [8].

En este proyecto de tesis, un prodigio del territorio se presenta como una narración acerca de las experiencias relacionadas a la preservación del territorio pese a las adversidades ambientales, sociales, climáticas y económicas. Dichas narrativas forman parte de una estrategia de comunicación que permite el debate sobre las particularidades del territorio entre los ciudadanos, la valoración y apropiación del entorno local, así como la reivindicación de iniciativas autogestivas para la conservación y restauración que los habitantes de un territorio realizan día con día.

1.1.9 Metodología de Prodigios del Territorio

Teniendo en cuenta la definición anterior, la Metodología de Prodigios del Territorio se define como un proyecto interdisciplinario que consiste en la observación, selección, narración y difusión, mediante espacios socio digitales, de los elementos socioambientales del entorno local que sorprenden a los habitantes de un territorio vivo. Inicialmente se le conocía como Metodología de Prodigios Ambientales, sin embargo, se consideró que los prodigios no necesariamente se limitan únicamente al aspecto natural de un territorio, sino que es posible que abarquen también el aspecto social, por lo que se incluyó dentro de la definición de la metodología tanto aspectos culturales, históricos, tradicionales, naturales y ambientales del territorio. El objetivo de la metodología consiste en el desarrollo de un espacio de diálogo e intercambio sobre las formas en las que se habita, preserva y restaura un territorio.

El proceso, identificado de la implementación de la metodología en [8], parte de la mano de un convocante, quien se encarga de invitar al público a la participación en actividades, formales o informales, para el reconocimiento del territorio y la posterior redacción de trabajos en los que cada participante describe sus experiencias, pensamientos y emociones con respecto a la realidad ambiental observada. Las actividades de campo incluyen la identificación de prodigios alrededor de la localidad del participante, la llamada a actividades de preservación y conservación del territorio, el recorrido por áreas naturales, entre otras. Posteriormente, los participantes, en este punto agentes generadores, envían sus trabajos al convocante para su presentación dentro del espacio.

Los trabajos soportados dentro del espacio suelen tratarse de textos acompañados por imágenes, videos, audios o dibujos relacionados a lo redactado. Cada trabajo recibido pasa por un proceso

de revisión en el que se evalúa la calidad del contenido enviado y se adecua para su exposición. La presentación de los trabajos suele llevarse a cabo por medio de una galería, virtual o física, o a través de medios de comunicación tradicionales como revistas, programas de televisión, entre otros. Al tratarse de un espacio de diálogo y aportación pública, es requerida la intervención de un mediador encargado de la gestión de las convocatorias, contenidos y discusiones que se llevan a cabo dentro del espacio.

1.1.10 *Storytelling*

Storytelling, conocida como narrativa o narración en español, es una herramienta de aprendizaje que ofrece al público una moraleja derivada de un suceso específico. Toma como base el uso del lenguaje, la emotividad, la comunicación y la construcción de elementos de una historia para construir emociones, promover puntos de vista variados y, en general, brindar una enseñanza.

La narración, desde un punto de vista educativo, es un elemento de comunicación efectivo como estrategia para la humanización del aprendizaje, pues permite la comprensión de la realidad a través de distintas perspectivas, así como establecer conexiones entre personas con los mismo intereses o características. Las historias narradas tocan las emociones de quienes las escuchan, lo cual genera un contraste con la exposición convencional, pues no logra concientizar de la misma forma. La propuesta de la narrativa como una herramienta de educación se fundamenta en la demanda educativa actual pues fomenta las características y valores de una identidad; promueve la valoración y apreciación de la cultura; motiva a la interacción y el ejercicio del pensamiento; e impulsa la acción del estudiante para adaptar lo aprendido a su entorno personal, laboral y social [9].

Dentro de la literatura disponible se encuentra, entre otros autores, a Kieran Egan, creador de la teoría de educación imaginativa. En dicha teoría, se describe cómo la imaginación y las emociones se combinan con el intelecto para conseguir el aprendizaje. En [10] Boullosa explica diversos ejemplos en donde la combinación mencionada anteriormente se presenta en forma de narrativas, no limitando el área de conocimiento a la literatura o historia exclusivamente. Dentro del mismo libro se cita a Harai [11] respecto a la idea de que el Homo Sapiens adjudica su permanencia y dominio el mundo (positivo o no) gracias a su capacidad de creer en conceptos

intangibles (como el dinero, el estado y los derechos humanos) en grupos numerosos, y cómo tales conceptos se transmitieron mediante relatos orales o narrativas.

1.1.11 Escritura funcional y creativa

La escritura creativa se define como aquella que crea historias utilizando formatos y esquemas que no son estándares dentro de la escritura literaria, periodística, técnica o académica [12]. Pese a carecer de normas y estándares establecidos, en la escritura creativa se identifican las siguientes características:

- Tiene como base la originalidad, pues es innovadora, diferente y aporta nuevas perspectivas.
- Posee una libertad total de creación, no se limita a los géneros tradicionales.
- Es transgresora, es decir, rompe las normas tradicionales en lo que a escritura se refiere.

Desde el siglo diecinueve surgió una línea de pensamiento que busca la rehabilitación del arte y la literatura como forma de conocimiento de la realidad. Debido esto, nace el interés por la enseñanza de la creación de este conocimiento en las aulas como camino para la investigación narrativa y, consecuentemente, surge la docencia en escritura creativa [13].

1.1.12 Participación ciudadana

Según el diccionario de la Real Academia Española, la participación es la acción y efecto de participar. Se trata de un proceso que incluye la posibilidad de tomar parte en algo con la finalidad de provocar una reacción. Por parte del ciudadano, este se define como una persona considerada como miembro activo de un estado, titular de derecho políticos y sometidos a sus leyes [14]. Con base en lo anterior, la participación ciudadana se entiende como el ejercicio de la expresión del derecho democrático y un proceso encargado de vigilar dicho ejercicio con el objetivo de que tal derecho no sea solo una ley en papel [15].

Es importante señalar que la participación ciudadana no está limitada únicamente al ejercicio del voto en procesos electorales, existen otros mecanismos que permiten la participación de la sociedad civil en la toma de decisiones, políticas o sociales, que no tienen que pasar necesariamente por los canales institucionales. En [16] se argumenta por el potencial de las tecnologías de la información y la comunicación como medio por el cual surgen nuevas

oportunidades para informar a la juventud y facilitar su participación, por medio del intercambio de información diversa gracias a su interactividad. A su vez, se hace referencia al establecimiento de centros voluntarios orientados al apoyo y promoción de la participación en actividades voluntarias, como campañas de información y promoción, así como el establecimiento de sistemas que reconozcan y valoren el trabajo voluntario como medio para fomentar principios fundamentales de la participación de los jóvenes en las decisiones y actividades locales y regionales.

1.1.13 Apropiación del Territorio

El término “Apropiación del Territorio” se define en [17] como: “un proceso dialéctico por el cual las personas y los espacios se vinculan dentro de un contexto sociocultural, desde un nivel individual, grupal o comunitario hasta un nivel social”. Lejos del sentido habitual de “apropiación”, asociado a la adquisición indebida de un bien, el objetivo de este proceso se encuentra delimitado, principalmente, por el concepto de apego al lugar. Dicho apego, dentro del modelo establecido en [17], se desarrolla gracias a la ayuda de 2 vías principales: la acción-transformación y la identificación simbólica.

Es a través de las acciones realizadas sobre el entorno que las personas y grupos dejan su propia marca simbólica en el territorio, lo que permite incorporar a este último en sus procesos afectivos y cognitivos de forma activa y constante. Gracias a estas acciones y la interacción con el territorio, ya sea individual o social, el espacio en el que se habita se nutre de un significado para sus habitantes. Por otro lado, la identificación simbólica permite tanto a grupos sociales como individuos el reconocer el terreno en el que se desenvuelven, atribuyendo cualidades de su entorno como parte de su identidad.

1.1.14 Aplicación Web

Una aplicación web se refiere a un tipo de aplicación a la que se accede por medio de un visualizador web a través una red y se desarrolla empleando lenguajes compatibles con el visualizador [18]. En este tipo de aplicación se identifican tres niveles: el nivel superior, de interacción con el usuario; el nivel inferior, que proporciona los datos; y el nivel intermedio, que se encarga del procesamiento de los datos.

Por lo regular, las aplicaciones web trabajan bajo una arquitectura cliente/servidor. Esta se divide en tres elementos:

- El cliente: representado por un programa a través del cual el usuario realiza solicitudes de recursos a un servidor. Por lo general, este programa está conformado por código HTML (Lenguaje de Marcado de Hipertexto) que se compila en el visualizador web junto a código ejecutable en lenguaje *script*.
- El servidor: hace referencia a un programa que, mientras se está ejecutando, espera perpetuamente las solicitudes de conexión hechas por los clientes. Por lo general, dichas solicitudes se realizan a través de un protocolo de comunicación. El lado del servidor se compone de una variedad de recursos como páginas (estáticas y dinámicas), repositorios de información y elementos adicionales como material multimedia o documentos.
- El Protocolo de comunicación: se refiere al medio por el cual los clientes realizan solicitudes al servidor. Por lo general, se emplea el protocolo HTTP (Protocolo de Transferencia de Hipertexto), que es parte de la familia de protocolos de comunicación TCP/IP empleados en internet [19].

1.1.15 Plataforma Colaborativa

Las plataformas colaborativas, conocidas también como *groupware*, son un sistema informático que soporta a grupos de personas comprometidas en una tarea u objetivo común y que proporcionan una interfaz para un entorno compartido [20]. El objetivo de dichos sistemas es el asistir a grupos de trabajo en la comunicación, colaboración y coordinación de sus actividades. Esta definición excluye los sistemas multiusuarios, pues es posible que los usuarios no compartan una actividad común. Además, el concepto no especifica la actividad simultánea de los usuarios, pues las plataformas colaborativas que permiten esta simultaneidad son llamadas plataformas colaborativas en tiempo real.

1.1.16 Aplicación de una sola página

Una aplicación de una sola Página (SPA por sus siglas en inglés) se trata de una aplicación accedida a través un visualizador web que no recarga la página mientras se ejecuta. Estas aplicaciones son capaces de integrarse con una gran cantidad de tecnologías de servidor pues,

debido a que gran cantidad de las funciones de la aplicación radican en el visualizador web, los requisitos del servidor se vuelven significativamente reducidos [21].

1.1.17 Arquitectura de múltiples capas

Se refiere a la división lógica de una aplicación en “capas”, pudiendo encontrarse en “niveles” separados o dentro del mismo, que son capaces de comunicarse entre ellas, así como con otros clientes y aplicaciones, independientemente de su ubicación física. En este sentido, el término “capa” hace referencia a la agrupación lógica de la funcionalidad y los componentes en una aplicación, mientras que “nivel” describe la distribución física de la funcionalidad.

La organización en capas permite diferenciar las distintas tareas realizadas por los componentes, lo que facilita la generación de un diseño que permite la reutilización de los mismos. De este modo, cada capa lógica posee una serie de componentes agrupados en subcapas, en la cuales se realiza un tipo tarea específica. Esto maximiza la mantenibilidad del código, optimiza el funcionamiento de la aplicación y proporciona una división de clara que facilita la toma de decisiones lógicas y de diseño.

El ejemplo más común de este tipo de arquitectura es el de tres capas, como se muestra en la figura 1.1. Este se compone por las capas de presentación, lógica del negocio y acceso a datos.

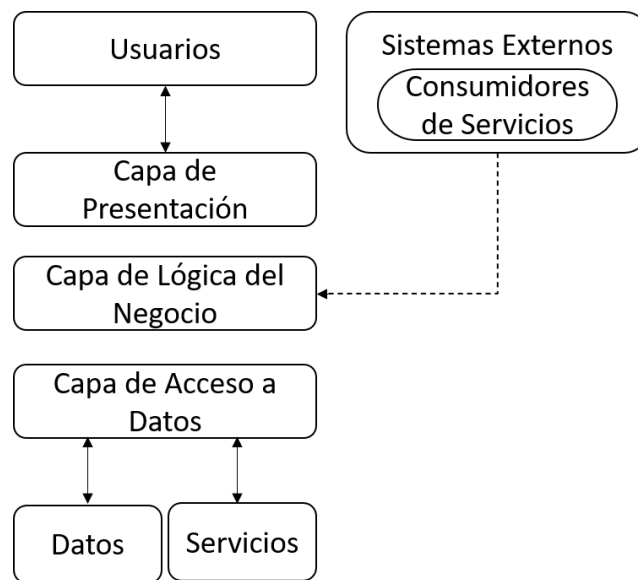


Figura 1.1 Ejemplo de arquitectura de múltiples capas

La capa de presentación alberga la funcionalidad dirigida al usuario que interactúa con el sistema, consistiendo principalmente en componentes que establecen un enlace con la capa de lógica del negocio. Esta última, por su parte, implementa la funcionalidad central del sistema, conteniendo la lógica relevante del negocio y presentándola en forma de componentes que pueden exponer interfases de servicio que son llamadas por la aplicación o, en ocasiones, por aplicaciones de terceros. Finalmente, la capa de acceso a datos, proporciona un medio a través del cual alcanzar los datos alojados dentro del sistema o bien, los datos expuestos por otros sistemas en red. Esto lo logra por medio de interfaces genéricas que los componentes de la capa de negocio pueden consumir.

1.2 Planteamiento del problema

En la actualidad, debido a las problemáticas ambientales que se experimentan y el agotamiento de las instituciones ante dichas situaciones, surgió la necesidad de fomentar una educación ambiental, no necesariamente formal, que promueva valores éticos, económicos y estéticos además de impulsar la participación ciudadana, individual y comunitaria, considerando al individuo como un actor capaz de lograr una transformación social de su entorno para generar una realidad ambiental distinta. Con este propósito, se requiere que la educación funcione como un proceso social, originado por un intercambio de conocimientos, vivencias, experiencias y simbolismos, que potencien el aprovechamiento de toda información para genera un impacto positivo en la realidad ambiental actual.

Gracias al surgimiento del paradigma de la educación como proceso reflexivo, la narración se ha incorporado como herramienta de aprendizaje dentro de diversas áreas de conocimiento, sirviendo como una estrategia de humanización del aprendizaje. Esta propuesta de innovación educativa encaja dentro de las demandas actuales de la educación ambiental pues no solo promueve los valores, la esencia y las características de una identidad, sino que también, gracias al elemento emocional que se emplea dentro de la propia narración, se promueve la interacción, el ejercicio del pensamiento, la valoración y apreciación de la cultura, y la motivación a la acción, dirigida al agente en formación, para adaptar el conocimiento obtenido a la experiencia, el trabajo, la vida y la sociedad.

Como ejemplos de implementación de esta modalidad educativa, en México, organizaciones no gubernamentales han realizado estrategias colegiadas, como la metodología de Prodigios Ambientales, que respondieron a los fenómenos y problemáticas socioambientales de regiones específicas, logrando generar un mayor compromiso hacia el territorio por parte de la ciudadanía. Pese al éxito producido, el alcance de estas estrategias se ve limitado a un nivel regional y los contenidos generados, si bien se encuentran publicados para su libre acceso, en la mayoría de las ocasiones se encuentran dispersos y/o pasan desapercibidos para el público general.

Por otro lado, con el auge de las nuevas tecnologías de comunicación y la digitalización del espacio social, la sociedad goza de un medio de comunicación instantáneo capaz de alcanzar a personas de todo el mundo que, entre otras ventajas, facilita la organización dentro y entre organizaciones colectivas, lo que permite el establecimiento de comunidades en línea y, por extensión, promover el dialogo, así como una cultura de participación entre los usuarios de dichos grupos. Es debido a estas nuevas situaciones que se dio el surgimiento de software de soporte para trabajo colaborativo, pues la organización dentro de un grupo no solo involucra el intercambio de mensajes, sino que abarca tanto el manejo de información, documentos y calendarización, por mencionar algunos ejemplos, y por lo tanto se requiere que la coordinación entre individuos se dé de la manera más natural y cómoda para los usuarios.

Tomando en cuenta lo anterior, se hace evidente la necesidad de un espacio digital que funja como sitio de recopilación y difusión centralizado de los contenidos afines a las necesidades de la educación ambiental actual y que, a su vez, funcione como un medio para el llamado a la participación de la ciudadanía a través de la escritura formal y creativa. En la literatura científica se encuentran descritos diversos proyectos que, pese a aplicarse en campos distintos, centran sus esfuerzos en la implementación de plataformas y herramientas colaborativas con el fin de facilitar la participación y el trabajo conjunto de los agentes interesados, así como el soporte a procesos pedagógicos y educativos. En la presente tesis, se planteó el desarrollo de un prototipo de plataforma colaborativa que da seguimiento a la metodología de “Prodigios Ambientales” por medio de la utilización de tecnologías web.

1.3 Objetivo general y específicos

En esta sección se definen los objetivos a obtener por parte del proyecto de tesis.

1.4.1 Objetivo General

Desarrollar el prototipo de una plataforma colaborativa para el seguimiento de la metodología de “Prodigios del Territorio”, a través de la utilización de tecnologías web.

1.4.2 Objetivos Específicos

1. Identificar los requerimientos funcionales y no funcionales de la metodología de “Prodigios Ambientales” para la determinación de los elementos que han de incluirse en el prototipo.
2. Diseñar la arquitectura tecnológica para soportar el prototipo y su eventual crecimiento como plataforma.
3. Realizar el diseño de sistemas del prototipo para cubrir los requerimientos identificados.
4. Desarrollar los elementos de código para integrarlos en el prototipo de la plataforma.
5. Probar el prototipo con, al menos, un caso de estudio para verificar su usabilidad.

1.4 Justificación

Según datos del Instituto Nacional de Estadística y Geografía (INEGI), la ecología y el medio ambiente son de las temáticas de participación y consulta, en espacios de administraciones públicas, con menor porcentaje de intervención del público, contando apenas con un 22.4% y un 23.2% respectivamente. La necesidad de que la sociedad civil participe en actividades del reconocimiento, preservación y apropiación del territorio es cada vez más imprescindible. Para cumplir este propósito, no es suficiente que la comunidad se organice de forma espontánea alrededor de temas de los que no está completamente convencida, sino que parta desde un pensamiento crítico, reflexivo y consciente de las problemáticas ambientales que experimenta su territorio en la actualidad.

Debido a las razones anteriormente mencionadas, es preciso incluir la escritura creativa y la educación ambiental dentro del marco de la participación ciudadana, pues la falta de información, o bien la participación desinformada, conlleva pérdida de derechos y, en el caso de lo planteado en este proyecto tesis, del territorio como espacio biocultural. En diversos planes

de estudio, tanto formales como no formales, la narración es poseedora de una relevancia como estrategia educativa, pues no solo se limita a la transmisión de conocimientos, sino que también tiene la capacidad humanizar el aprendizaje, permitiendo una mejor comprensión de los temas que toca y motivando al estudiante a implementar lo aprendido dentro de su entorno social. Es por medio de una mejor educación y de mejores espacios de discusión y difusión de conocimiento que la participación ciudadana se ve impulsada dentro de la sociedad civil.

Casos como el Laboratorio Socioambiental Ciudadano [8], que toman como estrategia la modalidad educativa planteada para la resignificación del territorio y sirven como espacio digital para el diálogo y la participación ciudadana, se ven limitados en su alcance debido a la falta de apoyo tecnológico y/o la falta de visibilidad. Gracias a la existencia de las tecnologías web, las organizaciones son capaces de ofrecer servicios, contenidos y entornos colaborativos a los que los usuarios acceden desde cualquier sitio de forma fácil e inmediata, pues no requieren de la instalación de pesadas aplicación o cumplir con requisitos exigentes para emplearlos, solo basta con el acceso a un visualizador web para hacer uso de dichas tecnologías. Por otro lado, el uso de estas tecnologías no tiene costos elevados, lo que las convierte en una opción atractiva para organizaciones que no disponen de muchos recursos.

Considerando lo establecido anteriormente, el presente proyecto resulta de gran importancia para las prácticas de educación ambiental y participación ciudadana, llevadas tanto por organizaciones gubernamentales y no gubernamentales, ya que ofrece un espacio centralizado, y de fácil acceso, para la aportación, discusión y difusión de contenidos formativos adecuados al modelo educativo requerido para dichas prácticas, pues parte del objetivo de la propuesta es dar soporte a un metodología que involucra el llamado a la participación de la sociedad civil por medio de la escritura funcional y creativa, y que genera contenidos que fomentan el interés y la resignificación del territorio gracias a la colaboración de la ciudadanía.

Capítulo 2 Estado de la práctica

En este capítulo se presenta un resumen de los artículos de investigación que presentan una relación, directa o indirecta, con el proyecto de tesis realizado.

2.1 Trabajos relacionados

En [22] se propone una plataforma participativa abierta al público para la planeación urbana empleando *webGIS*, una web que ofrece un mapa extenso para diversos fines técnicos y de la que el sistema toma información de entrada para ofrecer un entorno de mapeo con información visible. La base de datos de la plataforma es administrada por medio de una *API RESTful* escrita en PHP 7.3, mientras que la interfaz de usuario fue codificada usando *ReactJS* y *Redux*. La plataforma se creó con el propósito de resaltar los aspectos esenciales del espacio público y gestionar las fases de la participación ciudadana en la planificación urbana. Además, la plataforma exporta la información de la planificación llevada por medio de redes sociales para su aprobación por el público. A la fecha de publicación de publicación de [22], la plataforma se encontraba en desarrollo, con planes de publicarse en mayo de 2020, siendo piloteada en Atenas y Tesalónica.

En [23] se describe el diseño conceptual de una plataforma participativa, llamada *Há-Vita*, sobre la preservación de la biodiversidad y la naturaleza para los visitantes de un centro turístico. El diseño de la plataforma se basa en la presentación de contenido resaltante de entrevistas hechas a la comunidad local del centro turístico, en el que expresan su experiencia y conocimiento del lugar abarcando temas como la herencia cultural, la diversidad de especies animales y plantas de la región, así como los productos y artesanías locales.

En [24] se analizó la extensión lograda por plataformas que siguen una aproximación de colaboración abierta en sus políticas tecnológicas y de conocimiento, así como en sus prácticas. Se realizó un análisis empírico basado en 100 casos de Barcelona. Con base al análisis realizado, se concluyó que la modalidad abierta de las plataformas digitales colaborativas no está presente en la mayoría de los casos, existiendo diferentes áreas con distintos niveles de difusión de prácticas abiertas. El análisis, además, señaló la existencia de una correlación entre las políticas de tecnología, datos y conocimiento con los modelos democráticos de economía colaborativa,

lo que sugirió la importancia de las tecnologías y el conocimiento abierto en la adopción de modelos colaborativos democráticos y abiertos.

Por su parte, [25] exploró el rol de los medios colaborativos como herramienta para inspirar a la población hacia búsqueda de soluciones contra el cambio climático. Como caso de estudio se empleó el sitio de medios colaborativo *Youth4Climate*, en donde la gente publica, comenta y vota ideas para combatir el cambio climático. La investigación consistió en la identificación de los llamados a la acción encontrados dentro del proyecto, así como sus fases y los roles existentes en cada uno. A su vez, se estudiaron las herramientas que la plataforma proporcionó para la colaboración y los contenidos que produjo. El análisis de las publicaciones realizadas en el sitio permitió identificar ocho funciones sobre como los medios colaborativos apoyan las transformaciones de sustentabilidad ecológica impulsadas por la comunidad, contribuyendo a la compresión y estructuración de los medios colaborativos en el activismo ciudadano.

En [26] se evalúa el Sistema de Integración Comunal (SINCO), una plataforma que se desarrolló para el impulso del gobierno electrónico con el fin de generar más relaciones cooperativas entre el gobierno y los ciudadanos. La plataforma en cuestión comenzó como un sistema de información y administración de proyectos, sin embargo, evolucionó como un sistema de comunicación directa entre ciudadanos y gobernantes, dada las demandas de los usuarios. Para evaluar sus límites, la plataforma se sometió a un estudio de usabilidad. En su primera consulta se identificaron dificultades por parte de los usuarios debido a que la lógica con la que se desarrollaron los módulos no se ajustaba a la de estos. Para su segunda versión, la plataforma solucionó este problema, además de agregar otras mejoras como la integración de nuevos tipos de usuarios con el fin de lograr un mejor ejercicio de comunicación, responsabilidad con la información y control por parte de cada miembro. Al final de este análisis se determinó que existe una clara demanda por parte de los usuarios por cierta alfabetización tecnológica con el fin de lograr una apropiación más eficiente y precisa del sistema.

En [27] se identificaron los principales desafíos a los que se expone la participación en los procesos de “plataformización” en base a un estudio exploratorio del desarrollo de una plataforma digital de una municipalidad noruega. Estas dificultades se ilustran en tres puntos principales: la visión de los expertos con respecto a la participación a nivel municipal, los

mecanismos que restringen los espacios que posibilitan la participación y la posibilidad de escalar la plataforma. Como conclusión, se señala la necesidad de un marco de trabajo analítico para seguir los vínculos de decisión en el sector público, especialmente al evaluar el impacto de los factores de dificultad identificados en este trabajo.

En el artículo [28] se presentó el rediseño de *Participedia*, una plataforma de colaboración colectiva en línea diseñada para documentar y compartir conocimiento emergente sobre democracia participativa. El equipo de diseño y tecnología de *Participedia* empleó un diseño participativo junto con estrategias feministas de interacción humano-computadora para evaluar el sitio web, anteriormente empleado, y construir una plataforma de código abierto nueva. La implementación de la plataforma como un sistema de código abierto permitió la creación de oportunidades de participación para mujeres y estudiantes, e inició nuevas colaboraciones a través de canales como *Github*. Las mejoras del diseño, como la mejora de la accesibilidad y la reducción del sesgo en el modelo de datos, alinearon el proyecto a los valores deseados, la diversidad y la inclusión.

En [29] se planteó la importancia del aprendizaje electrónico, o *e-learning*, en la educación universitaria y el uso del modelado de información de construcción (*BIM* por sus siglas en inglés) como método didáctico para el mejoramiento de este tipo de aprendizaje. Como caso de estudio se implementó una herramienta *BIM* colaborativa en una clase de diseño de proyectos de ingeniería civil del Tecnológico de Monterrey. La herramienta proveyó un espacio de trabajo virtual centralizado eficiente para los estudiantes, mientras que el *BIM* proporcionó la suficiente capacidad y recursos para el modelado preciso de edificios. Los resultados obtenidos mostraron que la herramienta facilitó el intercambio de recursos dentro y entre los grupos de proyecto, reforzó las formas de colaboración grupal y mejoró los resultados del aprendizaje electrónico.

En [30] se explicó que, pese a que la agricultura es parte importante del tejido económico, existe una falta de personal especializado que eduque a quienes se encargan del sector agrícola. Como solución a esta situación, se propuso una plataforma colaborativa, conjunta con dispositivos de Internet de la Cosas, para el establecimiento de una comunicación en tiempo real entre granjeros y agrónomos en la que, mientras se mantuviese una videollamada, los agrónomos fueran capaces de diagnosticar eficientemente la situación gracias a la utilización de sensores colocados en los

campos. Tras verificar los resultados, se observó que la solución propuesta permitió a los ingenieros agrónomos una mayor eficiencia frente a la extensión de zonas por cubrir, pues la herramienta les permite intervenir de forma remota.

En el artículo [31] se estableció que, en los últimos años, el entrono computacional web *Jupyter notebook* se convirtió en la plataforma estándar de investigación colaborativa más popular entre los investigadores. Teniendo en cuenta este hecho, se presentó la primera implementación de *Sophon*, una plataforma web extensible para la investigación colaborativa basada en *JupyterLab*, que tuvo como objetivo extender la funcionalidad y mejorar la usabilidad de la herramienta sobre la que se desarrolló. La plataforma presentó diversas mejoras en comparación a su predecesora, como la capacidad de ajustar *JupyterLab* al flujo de trabajo de investigaciones preexistentes, el manejo de autenticación, el soporte a la colaboración de grupos de investigación y la investigación de información relacionada a los proyectos trabajados.

En [32] se investigó el cómo las plataformas de consumo colaborativo se diseñaron y evaluaron para mitigar el problema de la limitación de recursos en contexto de países en desarrollo. Con este propósito, se empleó el método de diseño de acción elaborada (*eADR*), junto con el paradigma de la ciencia de diseño, para el modelado y desarrollo de un artefacto digital de consumo colaborativo bajo el domino G2G (gobierno a gobierno). El producto obtenido se entregó a los usuarios de *WoredaNet*, quienes lo utilizaron para compartir, donar, alquilar e intercambiar activos físicos y digitales para aliviar el problema de limitación de recursos, lo que demostró su viabilidad como herramienta de optimización de recursos para organizaciones gubernamentales. En general, los resultados obtenidos contribuyeron al discurso de los sistemas de información en la práctica del gobierno digital, identificando la importancia de las plataformas digitales de consumo colaborativo para la optimización de los recursos gubernamentales en los países de bajos ingresos.

Por su parte, teniendo como base la importancia de la educación sobre la privacidad dentro de la vida de los adolescentes y su fuerte relación con la vida social de los mismos, [33] presentó la plataforma de aprendizaje colaborativo en línea *Teens-online*, una herramienta que proporcionó un entorno de aprendizaje colaborativo sobre los temas relacionados a la privacidad, al mismo tiempo que permitió a estudiantes compartir conocimientos, interactuar

con otros estudiantes, resolver exámenes de forma colaborativa, así como discutir situaciones y problemas relacionados con la privacidad. La plataforma se equipó con un mecanismo de emparejamiento basado en la teoría del juego de correspondencia, que permitió agrupar a estudiantes con las mismas características para cubrir mejor sus necesidades. Los resultados obtenidos de la implementación mostraron que la utilidad del mecanismo propuesto es mucho mayor que la obtenida por otros algoritmos. Además, se señaló que, gracias a dicho mecanismo, a los estudiantes se les emparejó con compañeros óptimos con los cuales fomentaron en mayor medida la participación en las actividades de aprendizaje.

En [34], con el fin de reducir la complejidad de la escritura de narrativas de forma colaborativa, se presentó *Cast*, una plataforma que permitió a un organizador definir de forma concisa el guion de una historia, que se implementó de manera oportuna para los usuarios por medio de una aplicación móvil consciente del contexto. El uso del contexto de los usuarios para definir sus roles proporcionó a los mismos las bases para promover un desarrollo narrativo cohesivo, y la automatización del proceso evitó la sobrecarga de trabajo del organizador. Los resultados de la implementación piloto demostraron que la creación de historias con la plataforma resultó atractiva para los participantes, pese a la dificultad que resultó la utilización de la plataforma para algunos usuarios.

En el artículo [35] se explicó que, pese a la gran cantidad de ecosistemas de “Todo como Servicio” en el mercado, existen pocos espacios dirigidos a la comunidad de investigación académica. Para solucionar esta situación, se desarrolló una plataforma web abierta dirigida a la comunidad científica a la que se le llamó *Agave*. Esta se enfocó en proporcionar interfaces y estándares fácilmente consumibles desde cualquier dispositivo, con el fin de ofrecer un ecosistema de “Ciencia como Servicio” para la ciencia reproducible. A la plataforma se le dotó la capacidad de ejecutar aplicaciones internas, llevar un sistema de almacenamiento de archivos, gestionar grupos de usuarios, enviar notificaciones, administrar cuentas de usuarios, etiquetar contenido de la plataforma, alojar inquilinos de producción (otras plataformas relacionadas a la investigación científica), entre otros. Como caso de estudio, el Centro de Computación Avanzada de Texas operó una versión empresarial de *Agave* y, para el año 2018, esta impulsó el uso de 10 inquilinos de producción para organizaciones en los Estados Unidos y Canadá.

Por su parte, en [36] se planteó que, debido al acelerado desarrollo de la urbanización y el nivel de industrialización China, la industria de la decoración se vio superada por problemas como la información opaca y la eficiencia industrial baja, lo que impidió establecer un buen orden industrial para el sector. Por otro lado, las compañías de decoración en línea tampoco lograron solventar dichos problemas efectivamente por lo que, con el fin de solucionar las problemáticas de la industria y promover su desarrollo, se propuso la creación de una plataforma de decoración colaborativa llamada *E-home*. En esta, los participantes emplearon modelos digitales basados en el modelado de información de construcción (*BIM*) como recurso intermedio, sobre el cual coordinaron su trabajo. La plataforma cumplió la función de emparejar equipos de trabajo conformados por clientes, diseñadores, equipos de construcción, proveedores de materiales y supervisores para solucionar los problemas generados por los grandes costos de coordinación de las compañías tradicionales, así como reducir los tiempos de selección del consumidor durante el proceso de decoración y, en general, mejorar la eficiencia de la industria de la decoración.

Pese a su surgimiento como tendencia tecnológica en el aprendizaje colaborativo, la colaboración en línea vio limitado su potencial debido la escasez de dispositivos colaborativos y la predominancia de herramientas de trabajo grupal para tareas específicas. Para tratar con esta situación, [37] propuso un enfoque de software colaborativo móvil, llamado *MobileMeeting*, con la capacidad ser integrado en las publicaciones de redes sociales que los estudiantes utilizan, como una herramienta para el aprendizaje colaborativo. Para validar la eficacia del enfoque propuesto, se realizó una investigación con casos reales basados en proyectos de estudiantes. El estudio confirmó que, empleando *MobileMeeting*, se mejoró las actividad y productividad del aprendizaje colaborativo, además de mostrar un incremento en la participación y retención de usuarios registrados, así como la introducción de nuevos.

Debido a la necesidad de habilidades informáticas y transversales, como la comunicación, la colaboración y la creatividad, que se requieren en la sociedad digital actual, y con el objetivo de crear una comunidad para los educadores de robótica que soporte la enseñanza en dicho campo, en [38] se desarrolló una plataforma de colaboración llamada *Roteco*, a la que se le dotó de: herramientas para el intercambio de experiencias entre participantes, cursos intensivos sobre los

fundamentos teóricos del campo y un repositorio de recursos educativos variados. Gracias a que la plataforma se desarrolló considerando las necesidades específicas de los educadores, se fomentó el desarrollo de las herramientas de apoyo a la enseñanza para los docentes, lo que generó un apoyo real a la comunidad de educadores.

Partiendo desde el cómo las tecnologías de la información y comunicación facilitaron la posibilidad de participar en actividades de recopilación de datos, y desde un punto de vista enfocado en cómo las tecnologías de la información y comunicación se emplean para incentivar a la ciudadanía a tomar un rol importante en el monitoreo y cuidado del ambiente, en [39] se estudió el caso del desarrollo de la plataforma *VattenFokus*. La plataforma funcionó principalmente como agregador de datos de ciencia ciudadana, herramienta de comunicación, sitio web y aplicación móvil para el monitoreo, recolección y difusión de información ambiental, centrando sus esfuerzos en la conservación de la calidad del agua en Suecia. Los resultados obtenidos de la implementación de la plataforma resaltaron que la integración de aplicaciones móviles y sistemas computacionales con propósitos de recopilación de datos ofreció una forma concreta de reunir actores (ciudadanos, funcionarios públicos, políticos, científicos, entre otros) que se interesan por su medio ambiente local, así como un espacio de dialogo entre ciudadanos y políticos para la deliberación de toma de decisiones y planeación colaborativa para actividades de conservación.

En [40], con el propósito de presentar una propuesta de negocio que permita a las empresas establecer sus propios flujos de trabajo colaborativo, se describió la plataforma participativa a escala mundial *Pl@ntNet*, un sistema dedicado a la monitorización de la biodiversidad vegetal a través de la identificación basada en imágenes. La propuesta se planteó como respuesta las demandas actuales de la agricultura y la economía ambiental, permitiendo a las empresas de estas áreas trabajar en su propio objeto de negocio con su propia comunidad de colaboradores y usuarios finales. La idea central consistió en el beneficio obtenido gracias a los avances del aprendizaje profundo y el aprendizaje transferido para mejoramiento del desempeño de tareas relacionadas con la identificación de plantas, como el reconocimiento de deficiencia de nutrientes, la patología o las causas de muerte de las plantas.

Con la aparición del modelo de participación colaborativa basada en web, la sociedad civil desarrolló un mayor interés por el proceso de participación electrónica, sin embargo, el proceso de participación ciudadana se vio desconectado del contexto de vida de los ciudadanos, lo que lo dotó de una falta de capacidad de respuesta. Con base a esta situación, [41] propuso un modelo conceptual para la contextualización de la opinión basado en las dimensiones semántica, espacial y temporal. Esta contextualización consistió la conexión de los aportes de los participantes con contextos relevantes que permitieron entender las problemáticas tratadas y aumentar la capacidad de respuesta del proceso de participación. El modelo propuesto se probó en una muestra aleatoria de información de transporte público en Canadá. El objetivo del estudio consistió en validar que la opinión ciudadana tiene una caracterización relevante con respecto a las tres dimensiones propuestas. Los resultados obtenidos demostraron que el 100% de los aportes de tienen una dimensión semántica, un 54% una dimensión temporal y un 67% una espacial.

Gracias a la digitalización y lo teléfonos móviles, los ciudadanos ganaron la capacidad de comunicarse más eficientemente con su gobierno, lo que condujo a una mayor democracia y transparencia en cuestiones relacionadas a la administración de su ciudad. Esta participación, llamada *E-participation*, se reconoció como un factor clave en el desarrollo de ciudades inteligentes. Con el fin de investigar el estado actual de las aplicaciones de participación en Alemania, en [42] se seleccionaron 29 aplicaciones que se categorizaron en cuatro temas: Conciencia Informativa, Servicio Municipal, Transparencia y Seguridad Ciudadana. El tema más común en la mayoría de las aplicaciones resultó ser “Servicio Municipal”, con el subtema de Identificación de problemas. Los resultados obtenidos del estudio demostraron que existen diferentes aplicaciones de participación desarrolladas por ciudad, por lo que resulta difícil para para los ciudadanos identificar la aplicación adecuada para su ciudad. Además, pese a la gran cantidad de desarrolladoras implicadas en este tipo de aplicaciones, no existen versiones de código abierto por lo que no existe información sobre el manejo de los datos.

Debido al surgimiento de sistemas de información que facilitan la participación ciudadana en la vida diaria de sus comunidades y la carencia de una comprensión sobre lo que permite la sustentabilidad en dichos sistemas, en [43] se introdujo un marco de trabajo para identificar y

analizar los factores que tienen influencia sobre la sustentabilidad de los sistemas de información locales. En este estudio se observaron treinta y cinco foros de discusión vecinal en línea y, con base a esta muestra, se analizó la relación entre las características de las comunidades en línea y la sustentabilidad en términos de atracción de nuevos usuarios, retención de usuarios existentes y desempeño de la producción de contenido. Los resultados evidenciaron que la sustentabilidad se relacionó significativamente con la demografía de sus comunidades objetivo, siendo la inestabilidad residencial de un vecindario el factor más importante. Los foros vecinales con mayor población transitoria son menos efectivos al atraer nuevos usuarios, fidelizar los actuales y generar nuevo contenido. Por otro lado, el tamaño de la población se relacionó positivamente con la atracción de nuevos usuarios, mientras que la diversidad racial con la retención y el desempeño.

En [44] se presenta el caso de estudio del desarrollo de *Alert4You*, un proyecto de medios comunitarios aplicado en 4 comunidades ubicadas en islas remotas de Santo Antão, en la Republica de Cabo Verde. El sistema constó de una mezcla entre tecnologías digitales y análogas, transmisión vía radio y una plataforma de información comunitaria. El aspecto colaborativo del proyecto se centró en la continua interacción de los actores involucrados, las decisiones tomadas sobre las tecnologías empleadas, la producción de contenidos para la radio y la plataforma, así como el marco de trabajo institucional. Tras el análisis al concepto e implementación del proyecto, se llegó a la conclusión de que las habilidades tecnológicas son importantes para la administración de plataformas digitales y lo equipos tecnológico empleados, así como la enseñanza del uso de la plataforma y la producción de audio. Como punto final, se señaló que la motivación por participar y el entusiasmo son factores igual de importantes, que pueden ser amenazados por las interrupciones surgidas durante el proceso, por lo que un flujo de trabajo institucional natural y sencillo es importante para la participación y la motivación. A futuro se planea investigar a profundidad cada comunidad involucrada con el fin de comparar las diferencias y similitudes de las plataformas comunitarias que se ejecutan en diferentes lugares en la misma isla.

En [45] se llevó a cabo una investigación sobre dos plataformas digitales enfocadas en la participación democrática: *Decide Madrid* y *Better Reykjavik*. El objetivo de este estudio fue

determinar si dichas plataformas son efectivas al ofrecer una mayor participación, así como una mejor toma de decisiones y, de lo contrario, proponer un nuevo modelo de participación en línea. Tras el análisis y la comparación de los dos casos de estudio, se identificaron los principales problemas compartidos por las plataformas: Generar conciencia y compromiso con la participación; generar ideas para solucionar problemas complejos; identificar propuestas de interés dada la gran cantidad de ideas, comentarios y sugerencias; recursos humanos limitados para administrar y revisar las ideas generadas por los usuarios. Con esto en cuenta, se desarrolló un marco de trabajo preliminar de factores para el surgimiento y la sustentabilidad de la inteligencia colectiva a nivel macro, cuyos resultados se decidieron utilizar para desarrollar un modelo para la participación en línea adaptado de los modelos de participación tradicionales.

2.2 Análisis comparativo

La Tabla 2.1 contiene la información de los artículos más relacionados con el proyecto de tesis con el fin de observar y comparar las semejanzas y diferencias de los atributos comunes encontrados cada uno.

Tabla 2.1 Análisis comparativo de los artículos revisados

Artículo	Problema	Contribución	Resultado	Estado
M. Karhu et al. [25]	Exploración de los medios colaborativos como herramienta para el empoderamiento de las personas hacia la sustentabilidad y la búsqueda de soluciones contra el cambio climático.	Informe sobre las estructuras y dinámicas de los medios colaborativos en el ámbito del problema tratado.	El trabajo ayuda a la comprensión de los medios colaborativos para el activismo ciudadano, trayendo un análisis y categorización de las funciones que ofrecen estas herramientas que permiten enmarcar y analizar el uso de esta herramienta en temas similares.	Terminada
B. Sánchez et al. [29]	Importancia del aprendizaje electrónico	Implementación de una herramienta de	Mayor facilidad en el intercambio de	Terminada

Artículo	Problema	Contribución	Resultado	Estado
	y la implementación de materiales didácticos para el apoyo de esta forma de aprendizaje.	modelado de información de construcción colaborativa en una clase de diseño de proyectos de ingeniería civil del Tecnológico de Monterrey	recursos dentro y entre grupos estudiantiles, reforzamiento de las formas de colaboración grupal y mejoramiento de los resultados del aprendizaje electrónico.	
O. Ly et al. [30]	La falta de expertos en ingeniería agrícola y la poca viabilidad del asesoramiento y diagnóstico de los campos.	Plataforma para la consulta y diagnóstico de tierras de cultivo solicitado a expertos por parte de los trabajadores del campo.	Mayor eficiencia en el proceso de asesoramiento y diagnóstico de los campos.	Terminada
R. Dooley et al. [35]	Ausencia de ecosistemas “todo como servicio” dirigidos a la comunidad de investigación académica.	Una plataforma “Ciencia como servicio” para la ciencia reproducible que brinda facilidades a los usuarios para llevar el proceso de investigación.	Potenciamiento de portales de producción de ciencia y mejora de la funcionalidad de otros servicios alojados.	Terminada
L. Negrini et al. [38]	Necesidad de la una comunidad que soporte la enseñanza en el campo de la robótica.	Plataforma de colaboración que provee un espacio para el intercambio de experiencias, cursos de la materia y un repositorio de recursos educativos variados.	Se generó un fomento significativo a las herramientas de apoyo a la educación	Terminada
T. Pargman et al. [39]	La forma en la que las tecnologías de la información y comunicación se emplean para incentivar	Análisis del desarrollo de una Plataforma para el monitoreo, recolección y difusión de información sobre la conservación de	Descripción y planteamiento de la influencia de las elecciones de diseño sobre la participación	Terminada

Artículo	Problema	Contribución	Resultado	Estado
	a la ciudadanía a tomar un rol en el monitoreo y cuidado del ambiente.	la calidad del agua en Suecia.	de los actores y los datos aportados por los mismo.	
A. Joly et al. [40]	Las demandas actuales de la agricultura y la economía ambiental, así como la necesidad de las empresas de estas áreas de trabajar en su propio objeto de negocio con su propia comunidad de colaboradores y usuarios finales.	Una plataforma participativa a escala mundial dedicada a la monitorización de la biodiversidad vegetal a través de la identificación basada en imágenes.	Una herramienta para la identificación de plantas y otras tareas como el reconocimiento de deficiencia de nutrientes y la patología o las causas de muerte de las plantas.	Terminada
C. López et al [43]	Carencia de una comprensión sobre lo que permite la sustentabilidad en sistemas de participación ciudadana.	Introducción de un marco de trabajo para la identificación y análisis de los factores que tienen influencia sobre la sustentabilidad de los sistemas de información locales enfocados a la participación ciudadana.	Se evidenció que la sustentabilidad de los sistemas se relaciona significativamente con la demografía de sus comunidades objetivo, siendo la inestabilidad residencial de un vecindario el factor más importante.	Terminado

Ante lo analizado en la Tabla 2.1, se llegó a la conclusión de que una plataforma de trabajo colaborativo es una solución viable para la resolución del problema planteado en este proyecto de tesis pues, como se observa en los artículos revisados, proporcionan un espacio que propicia y facilita la participación, colaboración, interacción y discusión entre los usuarios, así como una herramienta que facilita la generación e intercambio de recursos digitales.

Cabe resaltar que el presente proyecto complementa la parte tecnológica de la metodología de Prodigios Ambientales pues, pese a poseer un blog en el que se publican y difunden las actividades realizadas, carece de la estructura tecnológica necesaria para que los actores sean

capaces de participar a través del diálogo o la aportación de prodigios o de que existan convocantes que alienten la identificación y envío de materia para ser publicado.

Como principal contribución, este proyecto de tesis fomentará la escritura creativa, la participación ciudadana y el ejercicio de la práctica política, como se establece en los objetivos del Programa Nacional Estratégico (PRONACE) de Educación, gracias a que la plataforma establece la participación de los usuarios a través de la redacción de prodigios y comentarios hacia los contenidos presentados. Adicionalmente, aportará al reconocimiento a las diversas perspectivas y saberes, sociales y comunitarios, sobre la realidad ambiental del territorio de Veracruz, con el fin de incorporarlos en la consciencia del visitante de la plataforma.

2.3 Propuesta de Solución

En este trabajo se propone el desarrollo del prototipo de una plataforma colaborativa que se encargue de dar soporte a la metodología “Prodigios Ambientales”. La plataforma propuesta se encargará de permitir a diversos convocantes abrir espacios para el llamado a la redacción y la posterior recepción de narrativas de prodigios ambientales.

La arquitectura preliminar de la plataforma propuesta se muestra en la figura 3.1.

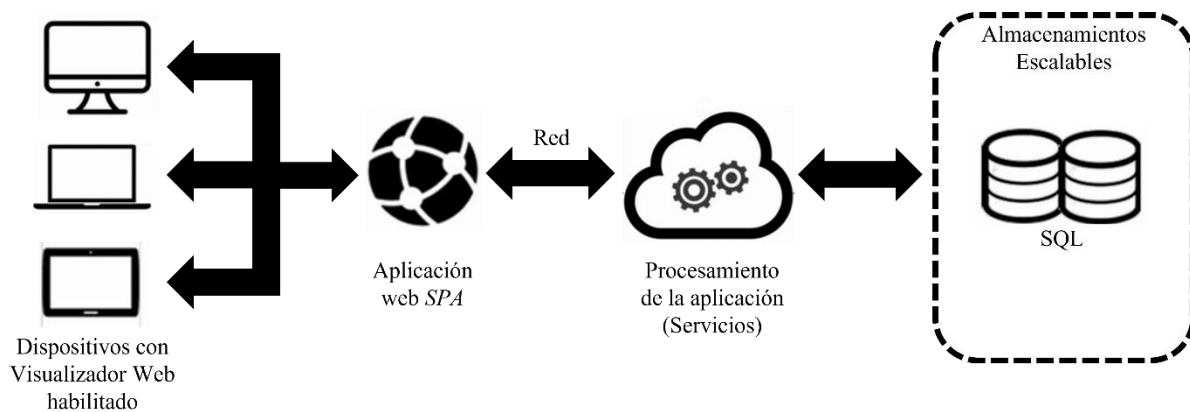


Figura 2.1 Arquitectura preliminar de la solución

La plataforma se encontrará disponible como una aplicación web de página única (*Single Page Application*, SPA por sus siglas en inglés), debido a la accesibilidad que ofrece al estar alojada en la web, la experiencia de trabajo fluida que presenta al usuario, y la gran simplicidad que conlleva su programación gracias a la considerable cantidad de marcos de trabajo orientados a

la creación de este tipo de aplicaciones. La primera aproximación del proceso esperado de la plataforma se describe en las figuras 3.2, 3.3 y 3.4.

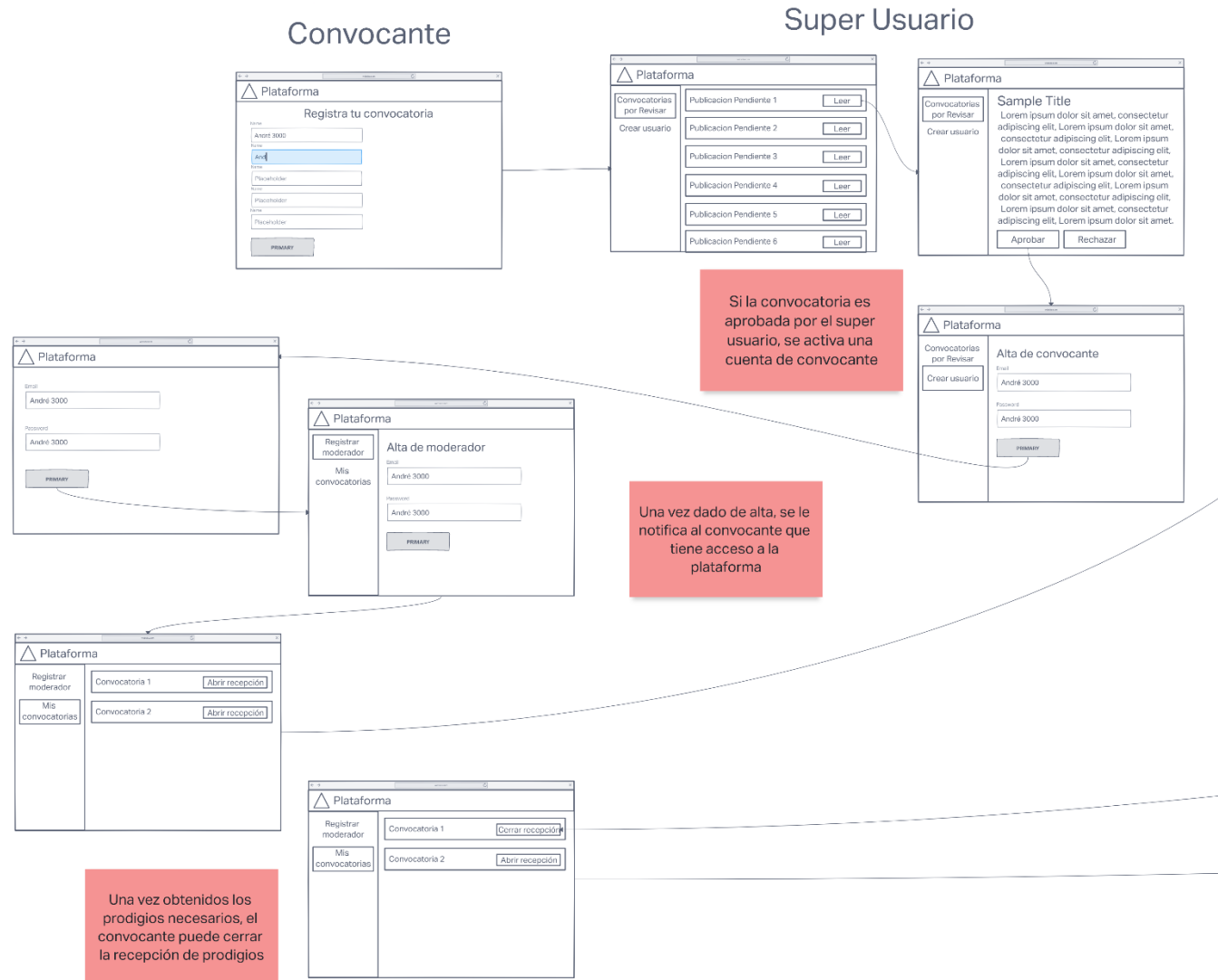


Figura 2.2 Bosquejo del funcionamiento de la plataforma (Parte 1)

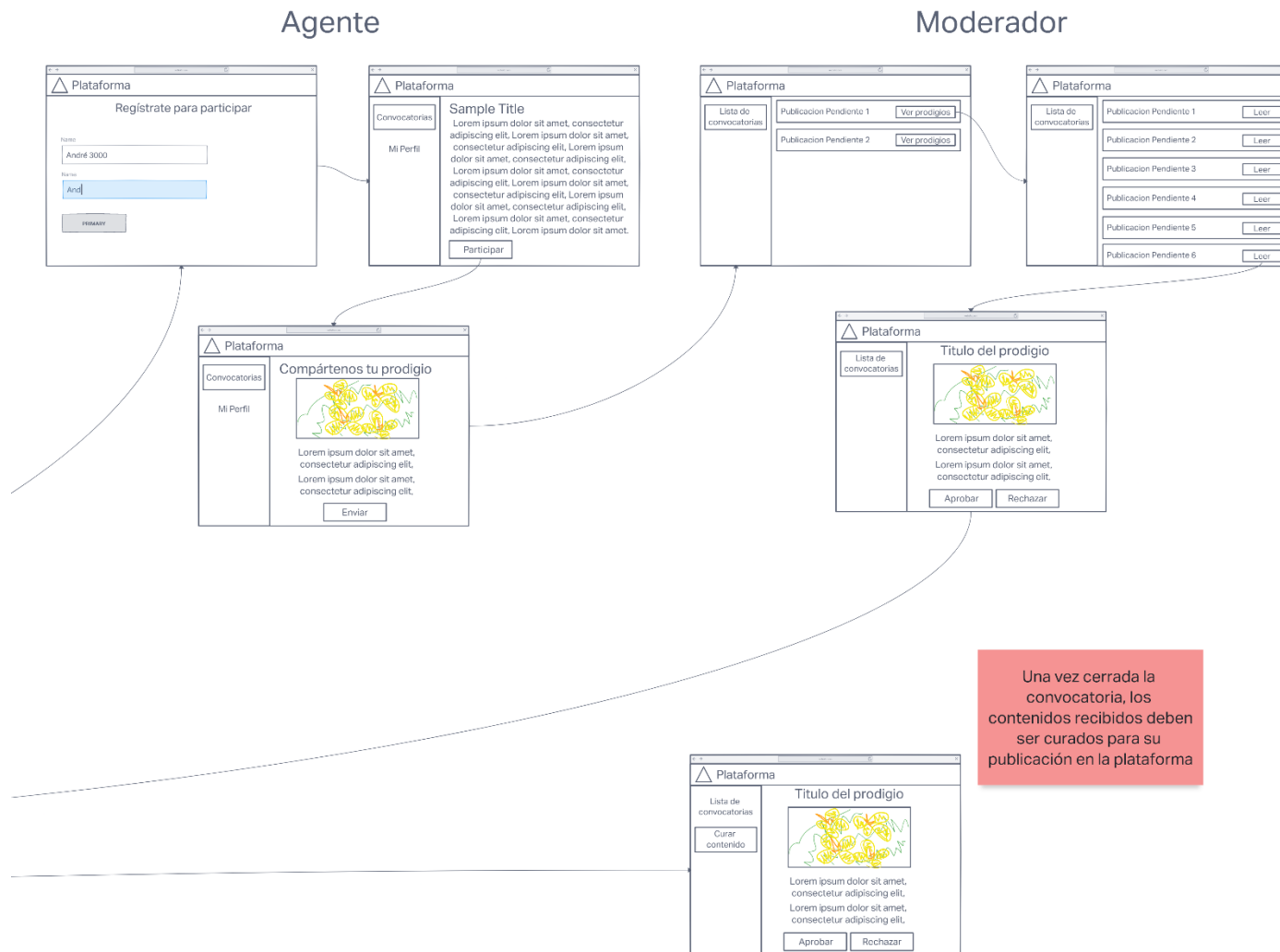


Figura 2.3 Bosquejo del funcionamiento de la plataforma (Parte 2)

Público General

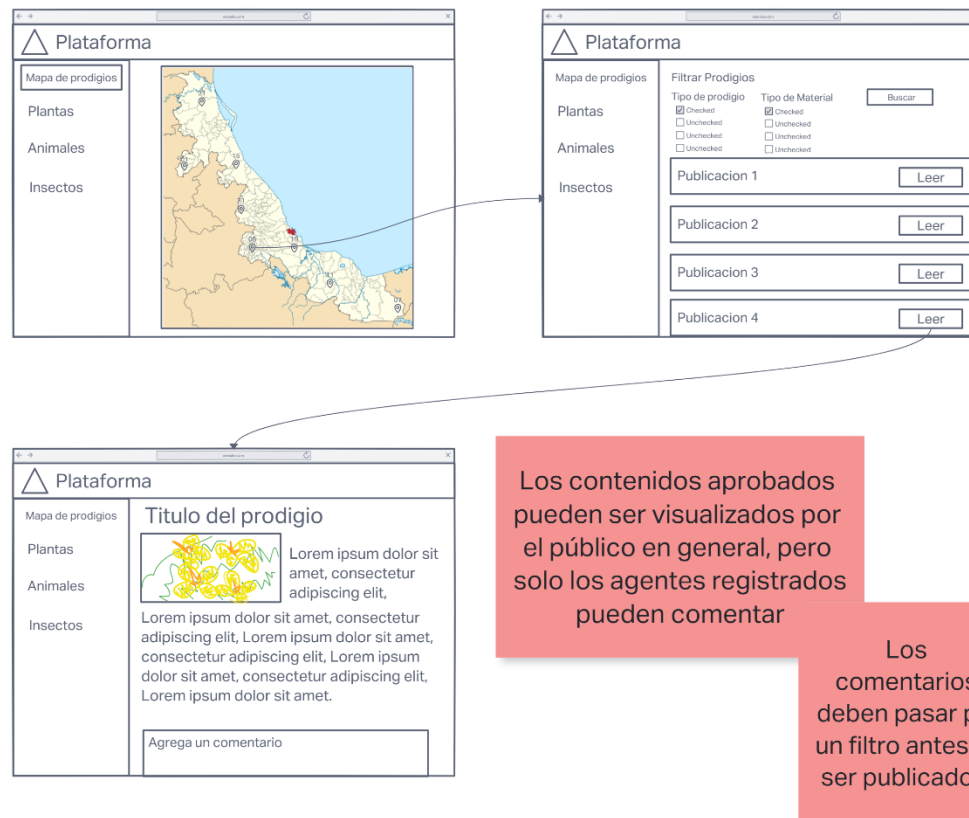


Figura 2.4 Bosquejo del funcionamiento de la plataforma (Parte 3)

Desde la aplicación, un convocante abrirá la recepción de prodigios ambientales a través de una convocatoria. Una vez abierta la recepción, los actores participantes inscribirán sus “prodigios” (texto, imágenes, videos, documentos o audios que narren la realidad ambiental del participante) para, posteriormente, ser aprobados y curados por un moderador, registrado previamente por el convocante, para su presentación dentro de la plataforma.

La plataforma almacenará los registros de convocantes, moderadores, convocatorias, actores y prodigios (un archivo multimedia y la información alrededor de dicho archivo) en una base de datos relacional, que será accedida a través de servicios web que se alojarán en el lado del servidor. Debido a que los prodigios que se subirán a la plataforma consisten en archivos multimedia, se optó por el almacenamiento de dichos archivos a través de cadenas binarias que se guardarán en la base de datos.

En primer lugar, para la codificación del *frontend*, se utiliza Angular como marco de trabajo para el desarrollo pues, además de ser el más maduro en comparación con los otros dos marcos de trabajo propuestos, no depende de la utilización de bibliotecas de terceros para la implementación de características como la solicitud de servicios, el enlace de datos o validación de formularios, ofreciendo un ecosistema que permite la creación de aplicaciones complejas sin requerir de mucho esfuerzo. Además, gracias a la herramienta de línea de comandos Angular CLI y de *widgets* complementarios, como *PrimeNg*, es posible la generación automática de código y elementos visuales, lo que agiliza y reduce los tiempos de codificación.

Para el desarrollo del *backend* se emplea el marco de trabajo Django, pues este otorga varias facilidades para la codificación de servicios REST (Transferencia de Estado Representacional) y es perfectamente aplicable tanto a proyectos pequeños como proyectos de gran complejidad. Adicionalmente, Django goza de utilidades que permiten mitigar diversos problemas de seguridad tales como inyecciones SQL, secuencia de comandos en sitios cruzados o falsificación de peticiones en sitios cruzados, así como una naturaleza versátil, una alta escalabilidad y un mapeo objeto relacional que facilita la consulta de base de datos.

Como base de datos se utiliza PostgreSQL debido a que es uno de los gestores de uso gratuito más populares, potente y robusto en la actualidad. Además de la gran cantidad de ventajas y

utilidades que ofrece PostgreSQL, este demostró un mejor desempeño y velocidad en conjuntos de datos de todos los tamaños, superando con creces a los otros gestores de bases de datos propuestos.

Para llevar el desarrollo general del proyecto de tesis, se empleó la metodología Scrum, ya que implica un proceso con etapas sencillas, permite la entrega de prototipos rápidos y un tiempo de planeación corto que agiliza el desarrollo. Debido a que Scrum es una metodología diseñada para equipos de desarrollo, para este proyecto de tesis se realizará una adaptación para una sola persona.

El despliegue de la plataforma colaborativa se realizó a través de los servicios de *Digital Ocean*, esto debido que ofrece las herramientas necesarias para alojar tanto el *frontend*, el *backend* y la base de datos, de modo que es posible centralizar el proyecto en una sola plataforma, lo que facilita su administración. Pese a ser un servicio de pago, los costos de *Digital Ocean* se consideran económicos en comparación a otras plataformas de alojamiento de aplicaciones web de paga.

2.3.1 Angular

Angular es uno de los marcos de trabajo más populares para la creación de aplicaciones web de una sola página. El marco emplea HTML y *TypeScript* para la codificación del lado del cliente. Las aplicaciones Angular se caracterizan por componerse de bloques de código básicos, conocidos dentro del marco como *NgModule*, que se encargan de recopilar código en conjuntos funcionales con el fin de darles un contexto de compilación para los componentes de la aplicación. Todas las aplicaciones poseen, por lo menos, un módulo raíz y múltiples módulos funcionales.

Cada componente de una aplicación Angular se conforma por una clase *TypeScript*, que contiene datos y lógica de aplicación, y una plantilla HTML asociada, que define la interfaz que mostrará al usuario. Adicionalmente, los componentes poseen la capacidad de emplear servicios “inyectados” como dependencias en los componentes, los cuales proveen de una funcionalidad específica no relacionada directamente con la interfaz. Este comportamiento asegura una estructura modular que permite la reutilización del código. Debido a la gran cantidad de vistas

que involucra una aplicación web, Angular ofrece un servicio de enrutamiento que permite definir las rutas de navegación entre las pantallas [46].

El marco de trabajo permite transformar, de forma rápida y sencilla, las plantillas diseñadas en código JavaScript optimizado para las máquinas virtuales modernas. Adicionalmente, no presenta problemas para acoplarse con aplicaciones de servidor desarrolladas en Node.js, .NET y PHP. Por su parte, las aplicaciones de Angular poseen una velocidad de carga considerablemente baja gracias a su sistema de enrutamiento de componentes, que permite cargar únicamente el código necesario para presentar la vista solicitada por el usuario [47].

2.3.2 Django

Django es un marco de trabajo de alto nivel basado en el lenguaje de programación *Python*. El marco de trabajo se enfoca en fomentar un diseño limpio y pragmático acompañado de un desarrollo rápido, con el cual, es posible llevar el proceso de construcción de aplicaciones web en cuestión de horas, desde su conceptualización hasta su despliegue. Entre las facilidades que ofrece Django, se encuentran herramientas para el manejo de tareas comunes como la autenticación de usuarios, la administración de contenidos y los mapas de sitio. Además, permiten a los desarrolladores a evitar muchos errores de seguridad comunes, como la inyección de SQL, las secuencias de comandos entre sitios, la falsificación de solicitudes entre sitios y el secuestro de clics [48].

2.3.3 PostgreSQL

PostgreSQL es un sistema de bases de datos de objetos relacionales de acceso gratuito, que posee una excelente reputación gracias a su arquitectura, confiabilidad, integridad y robustez. Entre sus principales ventajas, PostgreSQL se encuentra disponible para la mayoría de los sistemas operativos, y ofrece diversas características que permiten almacenar y escalar las cargas de trabajo más demandantes de forma segura [49].

En relación con este proyecto de tesis, el interés en PostgreSQL como posible base de datos de la plataforma es debido a que este sistema es capaz de almacenar cadenas binarias a través del tipo de dato *bytea*. Aunque el formato de entrada de *bytea* es diferente al que define el estándar SQL, BLOB, provee funciones y operadores que son esencialmente iguales. Esto resulta una

opción viable para el almacenamiento de archivos, pues es posible serializarlos con anterioridad (convertirlos a un formato de cadena binaria) para, posteriormente, ser registrados en un campo del tipo *bytea* [50].

2.3.4 *Digital Ocean*

Digital Ocean es una plataforma de hospedaje en la nube escalable, que ofrece diversos servicios para el alojamiento de aplicaciones web, *API backends*, base de datos, entre otros. *Digital Ocean* se encarga de rentar múltiples Servidores Privados en la Nube, llamados *Droplet*, que son administrados por enteramente por el usuario y permite a los desarrolladores centralizar sus proyectos en una sola plataforma. Los servicios de la plataforma son escalables, es decir, se paga por lo que se consume con la posibilidad de incluir más servicios conforme se requieran. En cuanto a precios, los costos de *Digital Ocean* son muy económicos en comparación con otras plataformas de alojamiento en el mercado (como el caso de *Azure* o *Amazon Web Service*).

La compilación y despliegue de las aplicaciones alojadas es llevada completamente por la plataforma, encargándose tanto de la infraestructura, dependencias y tiempos de ejecución de la aplicación mientras el desarrollador se enfoca completamente en el código. Los planes ofrecidos por la plataforma van desde los 5 dólares mensuales con una unidad central de procesamiento (CPU) y 512 MiB de memoria RAM, hasta los 12 dólares mensuales por un CPU y 1 GiB de memoria RAM. Por otro lado, las bases de datos soportadas por la plataforma incluyen a MySQL y PostgreSQL, con un costo de 15 dólares mensuales por 1 GB de memoria y 15 GB de almacenamiento en disco.

2.3.5 *Adaptación de Scrum*

Scrum es una metodología ágil diseñada para agregar energía, enfoque, claridad y transparencia a la planeación e implementación de proyectos. Esta metodología se estructura en ciclos de trabajos consecutivos conocidos como *Sprints* que, por lo regular, duran menos de un mes y son medidos en cuestión de semanas.

Al principio de cada ciclo, un equipo se encarga de escoger un conjunto de requerimientos de una lista priorizada y los construyen en su totalidad al final del ciclo. Durante este proceso, los requerimientos seleccionados no reciben cambios. Al final de cada ciclo, el equipo de desarrollo

hace una revisión de lo construido, así como la eficiencia de las prácticas empleadas. Posteriormente, los resultados se muestran a los interesados y, de este modo, se obtiene una retroalimentación que es posible incluir en el siguiente ciclo [51]. Cada *Sprint* se conforma de los siguientes pasos:

- Planeación del *Sprint*.
- Reunión Scrum Diaria.
- Revisión del *Sprint*.
- Retrospectiva del *Sprint*.

Debido a que Scrum es una metodología pensada para implementarse en equipos de desarrollo, el uso de Scrum en este proyecto de tesis será una adaptación adecuada para un único desarrollador.

Capítulo 3 Aplicación de la metodología

En este capítulo se presenta el desarrollo de la aplicación siguiendo la metodología indicada en el capítulo anterior.

3.1 Descripción de la solución

En el punto 2.3 del capítulo 2 se describe de forma general la solución a implementar. Esta consiste en una plataforma web formada por una arquitectura de tres capas: la de presentación, que se codificó empleando Angular, de lógica del negocio y la de acceso a datos, que se implementó a través de un servidor REST programado con Python, empleando Django REST Framework.

3.2 Metodología de desarrollo Scrum

Como se explora en el capítulo 2, la metodología Scrum se organiza en ciclos de trabajo conocidos como *Sprints*, durando cuanto mucho un mes, y se basa en una lista de funcionalidades a cubrir. Además, la metodología Scrum no considera iteraciones para arquitectura, sin embargo, en este caso sí se incluyen para facilitar el desarrollo. A continuación, se describen los *Sprints* realizados.

3.2.1 Sprint 1. Análisis

En esta etapa, durante el primer mes de realización de la estancia profesional, se obtuvieron los requisitos funcionales por parte de la codirectora del proyecto. Con base en estos requerimientos que se obtiene la información necesaria para la generación de los diagramas de casos de uso y clases. Es importante resaltar la dificultad que implica el diálogo entre los involucrados, pues surgen diversas problemáticas derivadas de la comprensión de ambos campos, técnico y social. Por ejemplo, desde el lado técnico, se busca delimitar el funcionamiento de la aplicación pensando en el proceso como una aplicación con alcance limitado, sin embargo, como se comprobó en posteriores sesiones, la Metodología de Prodigios del Territorio se encuentra en constante evolución. Además, la inclusión de actores sociales que no fueron considerados en un primer momento, como algunos pertenecientes al aspecto artístico y cultural, llevaron al cambio del foco del tema de tesis de Prodigios Ambientales a Prodigios del Territorio, lo que implicó un trabajo de escucha por parte del lado técnico al cual no se está acostumbrado.

Cabe señalar que, tras este primer *Sprint*, diversos aspectos de los diagramas en cuestión se modificaron en función de la información obtenida en posteriores reuniones con la codirectora y otros interesados en el proyecto, dando como resultado los diagramas presentados a continuación.

En la Figura 3.1 se presenta el diagrama general de casos de uso. Este se obtiene con base en los requisitos identificados y la narrativa del proceso de la metodología de prodigios del territorio. Los casos de uso mostrados se agruparon en función de los actores que se encargan de ejecutarlos siendo, a la fecha, seis los actores identificados para actuar con la plataforma. Cada uno de estos se describe a continuación.

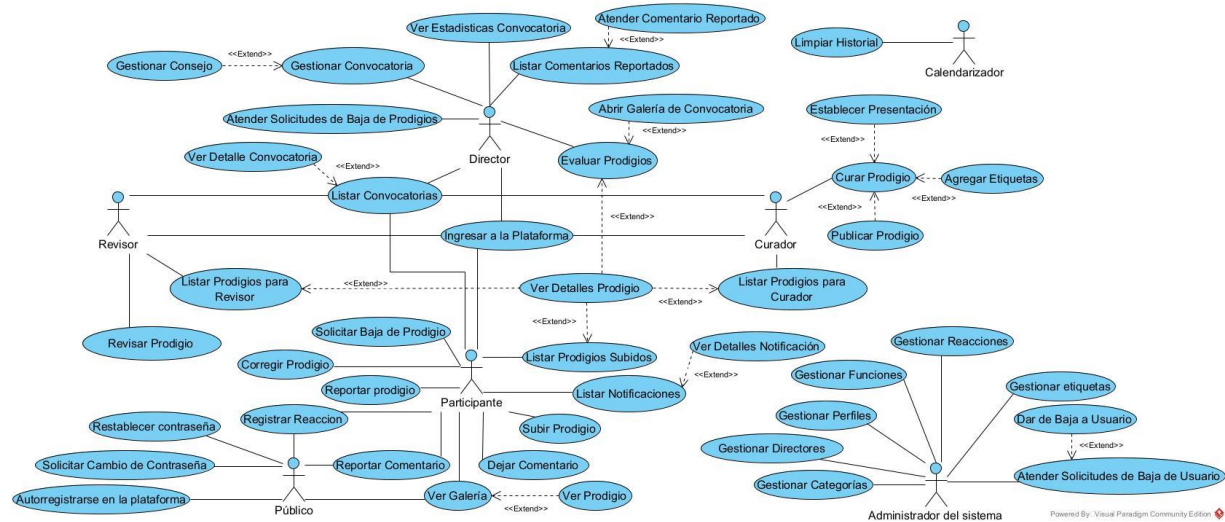


Figura 3.1 Diagrama de casos de uso general

Público: Hace referencia al actor que interactúa con la plataforma sin haberse identificado dentro del sistema. Como principales interacciones, posee la capacidad de revisar los prodigios publicados, crear una cuenta dentro de la plataforma (auto registro), solicitar y realizar el cambio de su contraseña de usuario, reportar comentarios y registrar reacciones sobre prodigios.

Participante: Se refiere al usuario básico identificado en la plataforma. Estos, además de compartir las funciones de registro de reacción, reporte de comentarios y visualización de la galería con el público, poseen tanto la facultad de subir prodigios a la plataforma, así como su consulta, corrección y solicitud de baja; son capaces de comentar, reportar en prodigios subidos

por otros participantes. Adicionalmente, poseen una bandeja de notificaciones con la cual podrán estar al tanto de acciones o acontecimientos relacionados con los trabajos de los cuales son autores.

Director: Es el responsable de la gestión de convocatorias y del Consejo encargado del proceso de revisión y curaduría. Cabe resaltar que, para la gestión de los miembros del consejo, esta se puede dar de dos formas: ya sea registrando un nuevo usuario que cumpla con alguno o ambos roles del consejo, o bien, dando a un usuario ya registrado (participante) permisos para ostentar alguno o ambos roles disponibles por lo que, al menos por el tiempo que el director determine pertinente, ganará una serie de funciones que se asocian a los roles asignados por el director.

Adicionalmente, el director posee permisos para atender las solicitudes de baja de prodigio, reportes de comentarios, el proceso de evaluación de los prodigios (exclusivamente aquellos subidos a sus convocatorias y que hayan sido inspeccionados por el equipo de revisión) y, finalmente, decide el momento en el las galerías generadas de una convocatoria se abrirá al público.

Revisor: Personal encargado del proceso de inspección; es un rol asignado a varias personas. Debido a esto, cuenta con permisos para la consulta de convocatorias y prodigios asociados. Adicionalmente, se encuentra facultado para registrar observaciones y calificaciones en los prodigios que revisa.

Curador: Facultado, junto con otros revisores, para llevar el proceso de curaduría de los prodigios. Entre sus tareas se encuentran el establecimiento de la presentación, el agregado de etiquetas y la publicación del prodigio (siendo esta última un cambio de estatus del prodigio a “lista para publicar”). Debido a lo anterior, y al igual que en el caso del revisor, el curador posee permisos para la consulta de convocatorias y prodigios asociados, específicamente aquellos que fueron aprobados por el director.

Administrador del Sistema: Principalmente, es el encargado de la gestión de recursos relacionados con el funcionamiento de la galería y el proceso general de la plataforma, como lo son las categorías, perfiles, funciones, reacciones y etiquetas. Además, es el único con permisos

para el registro de directores y la atención de solicitudes de baja de usuarios y su posterior eliminación.

Adicionalmente, se decidió anexar un **Calendarizador** que se encargará de realizar tareas básicas de forma automática tras ciertos periodos de tiempo o en fechas definidas, como es el caso de la eliminación de registros no utilizados o bien la apertura o cierre de convocatorias.

Como agregados, dentro del diagrama general, en un par de casos de uso, se generó un sub diagrama específico para cada uno que describe más a detalle el alcance de dichos casos de uso. Estos sub diagramas se explican a continuación.

La Figura 3.2 expande el caso de uso “Gestionar Consejo” segmentándolo en 5 casos de uso que, en su totalidad, cubren las operaciones sobre información almacenada (Lectura, Altas, Bajas y Modificaciones). Se agregó como sub diagrama debido a la complejidad que el diagrama principal ya presentaba.

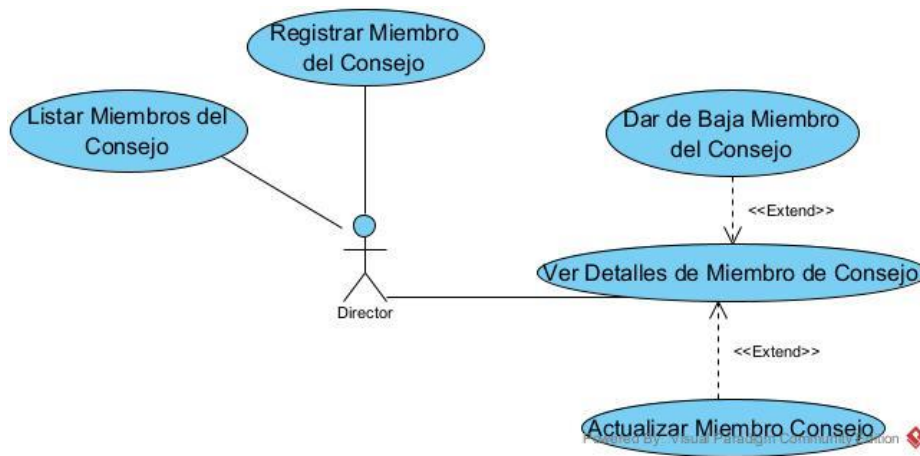


Figura 3.2 Sub diagrama de casos de uso, Gestionar Consejo

Por otro lado, la Figura 3.3 expande el caso de uso “Gestionar convocatorias”, indicando los casos de uso disponibles para el director y especificando los casos de uso de los que el **Calendarizador** se hará cargo.

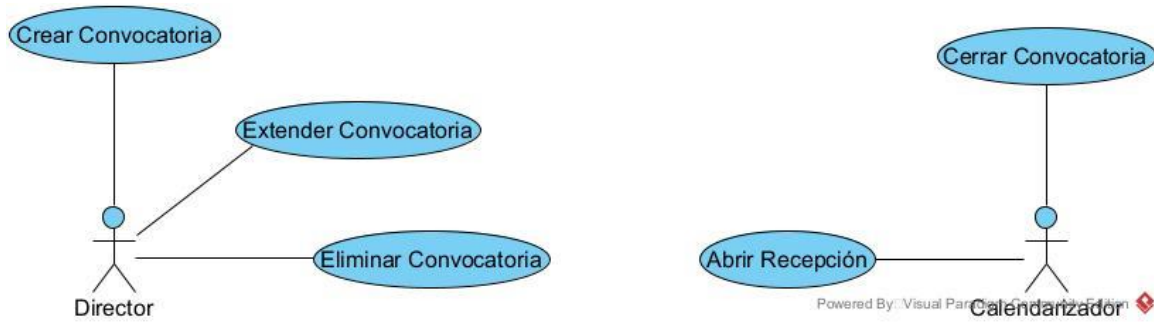


Figura 3.3 Sub diagrama de Casos de Uso, Gestionar Convocatorias

Posterior a la identificación de casos de uso, se realizó el diseño de las clases del sistema. Este se presenta en la Figura 3.4. El diseño de clases del sistema, al igual que el diagrama de casos de uso, fue obtenido a partir de los requerimientos funcionales y no funcionales y la narrativa del proceso de la metodología Prodigios Ambientales. Una breve descripción de las clases y las decisiones de diseños tomadas de describen en los siguientes párrafos.

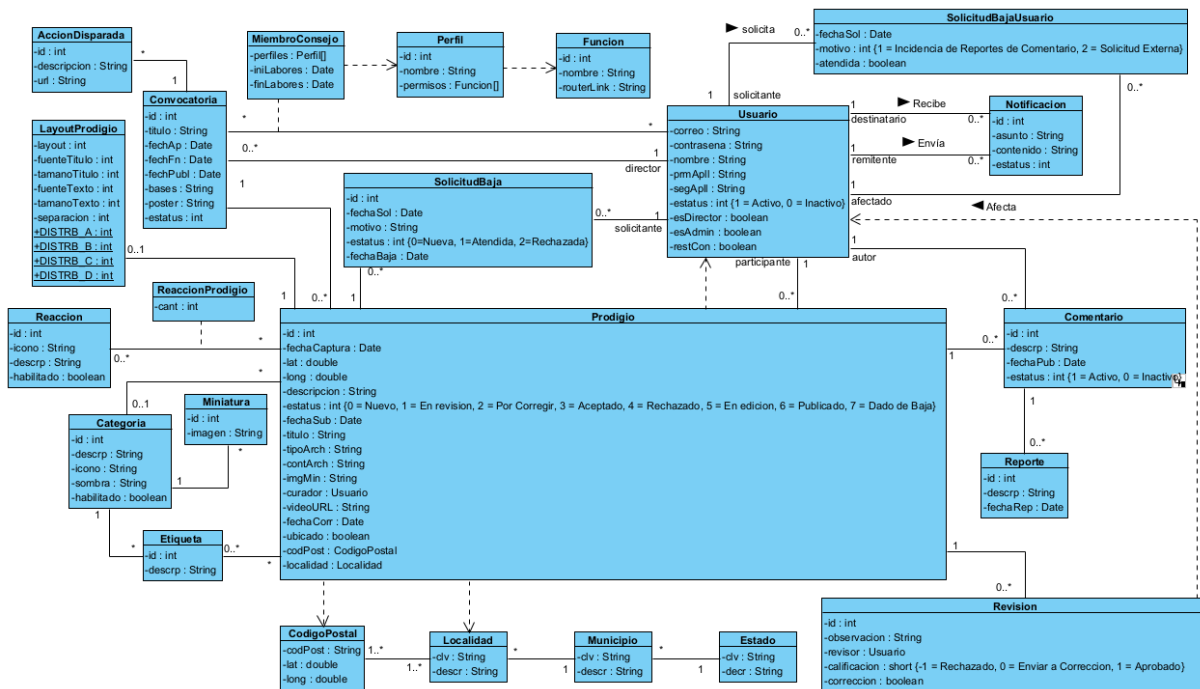


Figura 3.4 Diagrama de clases

Usuario: Tiene el propósito de representar a aquellos que posean una cuenta que los identifique en la plataforma. Posee un par de campos que permiten identificar a un usuario como Director o Administrador del sistema, esto debido a que estos roles de usuario garantizarán funciones

específicas de forma permanente y exclusivas para dichos roles, sin embargo, estos usuarios no requieren de más información.

Convocatoria: Describe las características que representa un llamado a la participación en la metodología de prodigios incluye para considerarse válido.

Categoría: Se refiere a la clasificación, por temática, en la que se asigna un prodigio. Cada categoría, a su vez, se asocia a una **Miniatura** específica (una imagen que representa de forma gráfica a la categoría) y una serie de **Etiquetas** que permiten precisar en mayor medida la temática del prodigio. Cabe resaltar que es posible asignar más de una etiqueta a cada prodigio, a diferencia de la categoría que sólo es una.

Estado, Municipio, Localidad y Código Postal son clases incluidas para la localización de prodigios por otro medio además de ubicación por latitud y longitud. De forma resumida, cada estado se conforma de una serie de municipios y estos, a su vez, se conforman de un conjunto de localidades. Finalmente, una localidad incluye uno o más códigos postales; los códigos postales tienen una latitud y longitud, lo que permite ubicar geográficamente al prodigio.

Prodigio: Representa el concepto central de la metodología de prodigios del territorio, se trata de los trabajos compartidos por los participantes dentro del espacio digital.

Revisión: Hace referencia a las observaciones realizadas por el equipo encargado de la inspección de los prodigios asociados a una convocatoria específica.

Layout Prodigio: Empleada durante el proceso de curaduría, representa las posibles distribuciones del contenido de un prodigio dentro de la pantalla, así como los tamaños de fuente con los que se presentará.

Solicitud Baja: Se refiere a las solicitudes de eliminación de prodigios de la plataforma, ya sea por petición del autor o por un tercero, siendo esta última, por lo regular, a causa de un reporte por infracción de derechos de autor.

Reacción: Hace referencia a una de las interacciones de los usuarios con los prodigios. Esta se visualiza, como en algunas redes sociales, con un emoticono que representa una emoción. Dado que un prodigio puede contar con diversas reacciones en cantidades distintas, y estas últimas se

encuentran en diversos prodigios, se decidió crear la agregar la clase de asociación **ReaccionProdigio**.

Comentario: Representa la opinión del participante respecto al contenido publicado.

Reporte: Si se da el caso de que un comentario infrinja las normas de comportamiento de la plataforma, cualquier participante (independientemente de su perfil) tiene la posibilidad de levantar un reporte, mismo que será atendido por el convocante/director (si se comprueba que infringe las normas, lo elimina).

SolicitudBajaUsuario: Dentro de la plataforma, si son comunes los casos en los que un usuario recibe reportes de comentario por conductas inadecuadas, un director solicitará la baja de un usuario aportando la fecha de solicitud y una explicación del motivo de la decisión.

Notificación: Son los avisos que se envían a los participantes para informar si el prodigio requiere correcciones o si sus comentarios han recibido reportes entre otros casos.

Perfil: Como se mencionó en el primer Sprint, los roles de **Revisor** y **Curador** son asignados tanto a usuarios ya existentes como a usuarios registrados para ese rol y solo durante un periodo de tiempo delimitado en una convocatoria específica. Es por esta razón que, buscando cubrir este aspecto de duración temporal y previendo que puedan aparecer más roles que actúen en la metodología, se decidió modelar estos roles en una sola clase **Perfil** de modo que, al asignarse a un usuario como Revisor o Curador este juegue uno o ambos Perfiles a través de la clase de asociación **MiembroConsejo**, que define tanto la convocatoria como el tiempo por el cual el usuario ostentará dichos perfiles.

Funciones: Hace referencia a las acciones que los perfiles de Revisor y Curador realizan en la plataforma.

Acción Disparada: Clase que representa las acciones derivadas de las convocatorias realizadas por los directores, por ejemplo, campañas de concientización, jornadas de limpieza, entre otras. Si bien es deseable que las convocatorias deriven en acciones que lleven al cambio positivo, estas no siempre serán conocidas por el director y/o no son soportadas por la plataforma, por lo que solo se registran siempre y cuando el convocante se entere de éstas.

3.2.2 Sprint 2. Diseño de la Arquitectura

El concepto general de arquitectura del sistema seleccionada para este proyecto se describió en la sección 1.1.17 y en la presente sección, detalla un poco más a fondo esta arquitectura. En primer lugar, se decidió emplear un modelo simple de tres capas: presentación, lógica de negocio y modelo de datos; representadas por la aplicación Angular, la Web API construida con Django REST Framework, un marco de trabajo basado en Python, y los modelos programados en Python respectivamente. Cabe resaltar que, si bien el marco de trabajo seleccionado para la codificación del servidor hace referencia a servicios REST (Django REST Framework), la API que se desarrolló no cumple con las características suficientes para ser considerada RESTful pues, en [52], se especifica que el termino REST en el nombre es únicamente empleado para la fácil identificación del marco de trabajo y no cumple en su totalidad con las especificaciones establecidas por Roy Fielding, creador del estilo arquitectónico REST.

Con el fin de satisfacer las necesidades relacionadas con funciones específicas que se buscan cubrir con la plataforma, a ciertas las capas de la arquitectura se anexaron algunas bibliotecas complementarias que permiten llevar a cabo tareas que, por sí solos, los elementos listados anteriormente no satisfacen fácilmente. Las decisiones tomadas en cuanto a estas bibliotecas adicionales se describen a continuación.

3.2.2.1 Mapas en la plataforma

Para la presentación de prodigios a lo largo del territorio se optó por emplear un servicio que permitiese generar mapas y ubicar puntos dentro de los mismos de forma dinámica no limitada (no se conocen de antemano las ubicaciones de los futuros prodigios), así como actualizar o sustituir dichos puntos. Como primera opción, se consideró la API de *Google Maps* disponible para Angular, sin embargo, debido a la fuerte inversión que implica el uso de este servicio para el caso de puntos dinámicos y la poca cantidad de solicitudes que se permiten realizar en relación a su precio, se decidió descartar esta alternativa. Posteriormente se optó por la API *Position Stack*, sin embargo, las funcionalidades requeridas por la plataforma se encuentran exclusivas para los planes con mayor costo por lo que se descartó como posibilidad. Finalmente, se consideró la biblioteca *leaflet* que, si bien es menos completa que las APIs anteriormente mencionadas, ofrece la funcionalidad básica que se busca para este proyecto y es gratuita.

3.2.2.2 Manejo de archivos en la plataforma

Como se explicó en la sección 2.3.3 se propuso el uso de cadenas binarias para el almacenamiento de archivos en la base de datos. Para la prueba de concepto, se decidió realizar este proceso de conversión a cadena con distintos tipos de archivos (los que se esperan recibir en la plataforma: jpg, mp3, mp4 y pdf) para medir los tiempos de conversión de cada uno y confirmar si este proceso era viable dentro del flujo normal de trabajo esperado en la plataforma. Como resultado, se notó que la conversión a cadena de archivos mp4 es la que demora mucho más tiempo en relación a los otros tipos e, inclusive, en el caso de alcanzar el tamaño máximo establecido, genera un error durante la conversión. Los tiempos de conversión identificados junto con los tamaños de prueba empleados se observa en la tabla 3.1,

Tabla 3.1 Comparación de tamaños de archivo y tiempos de conversión

Tipo/Formato de Archivo	Tamaño Máximo Establecido	Tamaño de la prueba	Tiempo de conversión a Cadena	Tiempo de conversión de cadena a Archivo
Audio (MP3)	5 min., 96 kbps, 4 Mb	5.1 Mb	24 seg.	12 seg.
Video (MP4)	5 min., Resolución HD 720p, 75 Mb	11.7 Mb	1:25 min.	3:07 min.
Imagen (JPG)	500 Kb	4 Mb	2 seg.	4 seg.
Archivo (PDF)	1 Mb	978 Kb	4 seg.	6 seg.

Debido a estos resultados, se buscaron otras opciones para el caso particular de videos. Como primera consideración, se planteó el uso de la API de *Youtube* para la carga y almacenamiento de videos y, para su posterior reproducción, se emplearía el reproductor embebido con el identificador que genera *Youtube*, sin embargo, al reducido número de cargas de video diarias permitidas (aproximadamente 6) se consideró que, considerando la afluencia esperada de la plataforma, la API no satisface completamente esta necesidad. Como segunda opción, se propuso emplear otra API, esta vez la de *Google Drive*. Esta, en comparación a la de *Youtube*,

ofrece un número mayor de cargas al día, suficiente para los picos esperados en las convocatorias.

En las pruebas de concepto se requirió de la biblioteca *pydrive* de Python como cliente de la API de Google Drive y un proceso manual para configurar las credenciales e identificación en la plataforma, con el fin de no se requiera cada vez que se utilice el servicio. Tomando en cuenta lo descrito anteriormente, el diagrama que representa la arquitectura final del sistema se muestra en la Figura 3.5.

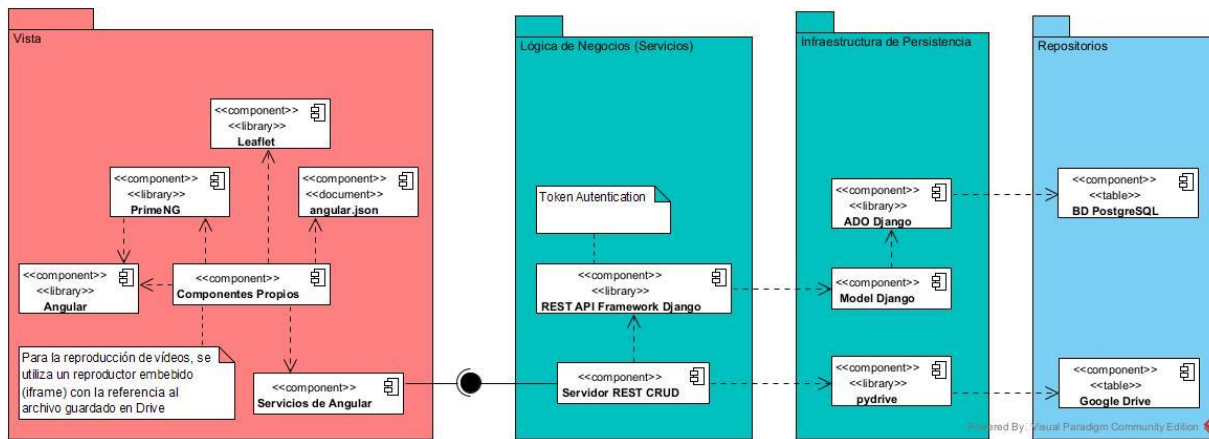


Figura 3.5 Arquitectura Final del Sistema (Diagrama de Componentes)

3.2.3 Sprint 3. Diseño de la aplicación

El diseño de la aplicación se centró en tres aspectos principales: diseño de las pantallas de la aplicación Angular, diseño de los servicios del *Web API* y diseño de la base de datos. Cada uno de estos se explica en los siguientes párrafos.

3.2.3.1 Diseño de las pantallas

Las pantallas de la aplicación se diseñaron como maquetados empleando la versión gratuita de la herramienta Balsamiq Cloud. Un pequeño ejemplo de este diseño se observa en la Figura 3.6, en la que se muestra la pantalla de inicio de la aplicación.

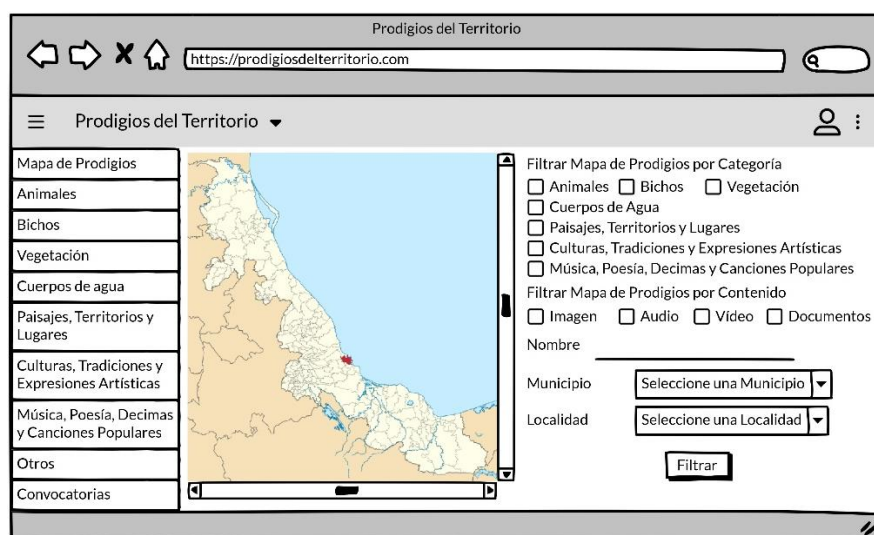


Figura 3.6 Pantalla principal de la plataforma

Cabe mencionar que parte de los maquetados fueron codificados en Angular, en un pequeño prototipo no funcional, con el fin de realizar una prueba de concepto del diseño de las pantallas, como se muestra en la Figura 3.7.

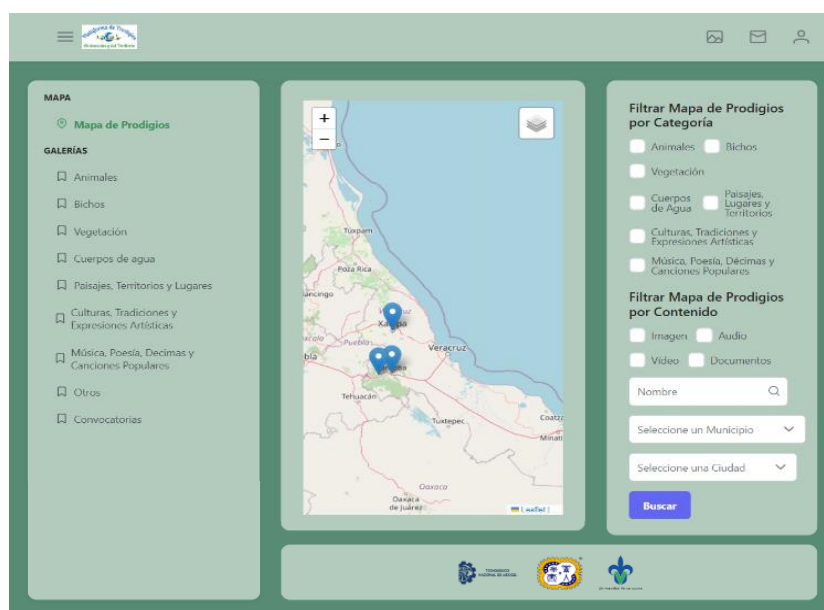


Figura 3.7 Pantalla principal codificada en Angular

La prueba de concepto se realizó en la Benemérita Escuela Normal Veracruzana Enrique C. Rebsamen, en Xalapa, Veracruz. Es importante destacar la dificultad que tuvieron los implicados en esta actividad para comprender el funcionamiento de los elementos de las

pantallas, así como para la identificación del proceso metodológico que el prototipo buscaba simular respecto a la plataforma. Los comentarios de los participantes se recolectaron mediante una encuesta y sirvieron para mejorar las pantallas, sobre todo en lo que respecta a la usabilidad. También es importante mencionar que los participantes no conocían a profundidad la metodología (como se espera que suceda con los subsiguientes participantes) pero sí tenían conocimiento sobre el uso de tecnologías web en general. Con base en los resultados de la encuesta se realizarán cambios en los diseños de las pantallas para asegurar el uso intuitivo de los menús y pantallas, así como una mejora visual en cuanto al uso de colores.

3.2.3.2 Autenticación de usuarios dentro de la aplicación

Para llevar las operaciones de inicio, cierre y administración de las sesiones de usuario se decidió utilizar el sistema proporcionado por el marco de trabajo elegido para trabajar, *Django REST Framework*. Específicamente, se emplea la autenticación por medio de *tokens* la cual, además de verificar la existencia de un usuario en la base de datos, genera un “*token*” en la base de datos que permite reconocer la sesión que un determinado usuario haya iniciado. Para esto, el marco de trabajo genera una serie de tablas propias que permiten soportar esta infraestructura.

3.2.3.3 Diseño de servicios

Para el diseño de servicios de la aplicación, se consideró cada narrativa de caso de uso y, con base en los procesos descritos de cada uno, se diseñaron los servicios necesarios para satisfacer el caso de uso. Cabe mencionar que existen servicios que se utilizan en más de un caso de uso y, pese a que cada servicio se encuentra asociado con un objeto en específico, en muchos casos la respuesta del servicio no será un objeto completo, sino que se obtendrán objetos parciales. Es debido a esto último que se implementó el uso de objetos de transferencia de datos (DTO por sus siglas en inglés) para gestionar de mejor manera la estructura del objeto de respuesta que retornará cada servicio.

En la Tabla 3.2 se muestra, a manera de ejemplo, el diseño de los servicios para el caso de uso Ver Galería. Es importante mencionar que, además de los parámetros de entrada y el JSON de salida de los servicios, se indica el uso de procesos almacenados en la base de datos para operaciones, ya sea de registro o consulta, que requieran de pasos adicionales como es el caso de aquéllos que requieren una conversión adicional a cadenas binarias para el almacenamiento de archivos en la base de datos.

Tabla 3.2 Diseño de los servicios del Caso de uso “Ver Galería”

Caso de Uso	ID	Recurso	Método	URL (app/apiprod/)	Parámetro GET/DELETE -> Query String POST/PUT -> JSON	JSON Salida (Formato de Fecha AAAA/MM/DD)	Código HTTP		Notas	Proceso almacenado
Ver Galería	1	prodigio	GET	prodigios/mapa	nombre, municipio, localidad, cveCat[], tipoMIME[]	[{ id:number, lat:number long:number nombre:String categoria:{ id:String, descrip:String, icono:String, sombra:String } }]	200	Ok		
							404	Not Found		
	2	prodigio	GET	prodigios/galeria	tipoMIME[], cveEtiqueta[], nombre, cveCat, cveConvocatoria	[{ id:num, nombre:String, imgMin:String }]	200	Ok		Se requiere proceso para convertir de binario a base 64
							404	Not Found		

Caso de Uso	ID	Recurso	Método	URL (app/apiprod/)	Parámetro GET/DELETE -> Query String POST/PUT -> JSON	JSON Salida (Formato de Fecha AAAA/MM/DD)	Código HTTP		Notas	Proceso almacenado
	30	etiqueta	GET	etiquetas/porCategoria	idCategoria	[{ Arreglo de Objetos Etiqueta }]	200	Ok	Si el id de la categoría es 0, obtén todas.	
							404	Not Found		
	3	convocatoria	GET	convocatorias/galería/recientes		[{ id:num, titulo:String, poster:String }]	200	Ok	Rango de búsqueda por definir Busca únicamente las convocatorias con estatus Galería Abierta	Se requiere proceso para convertir de binario a base 64
							404	Not Found		
	4	convocatoria	GET	convocatorias/porFiltro	idDirector idRevisor idCurador galería: boolean	Arreglo de Convocatoria	200	Ok	Si los 3 parámetros van vacíos, trae todas sin condiciones. Di el id es de director trae todas las convocatorias	
							404	Not Found		

Caso de Uso	ID	Recurso	Método	URL (app/apiprod/)	Parámetro GET/DELETE -> Query String POST/PUT -> JSON	JSON Salida (Formato de Fecha AAAA/MM/DD)	Código HTTP		Notas	Proceso almacenado
									registradas por el director Si el id es de revisor, trae las convocatorias en las que participa como Revisor Si el id es de curador, trae las convocatorias en la que participa como Curador Si es galería, busca únicamente las convocatorias con estatus	

Caso de Uso	ID	Recurso	Método	URL (app/apiprod/)	Parámetro GET/DELETE -> Query String POST/PUT -> JSON	JSON Salida (Formato de Fecha AAAA/MM/DD)	Código HTTP		Notas	Proceso almacenado
									Galería Abierta	

El resto de servicios están disponibles para su consulta en el archivo anexo “Servicios de la Plataforma”.

3.2.3.4 Diseño de la Base de Datos.

Ya con el diagrama de clases completamente diseñado y las funciones almacenadas previamente definidas, el diagrama de estructura física de la base de dato se realizó siguiendo los modelados previamente mencionados. La Figura 3.8 ilustra el primer diseño de la base de datos.

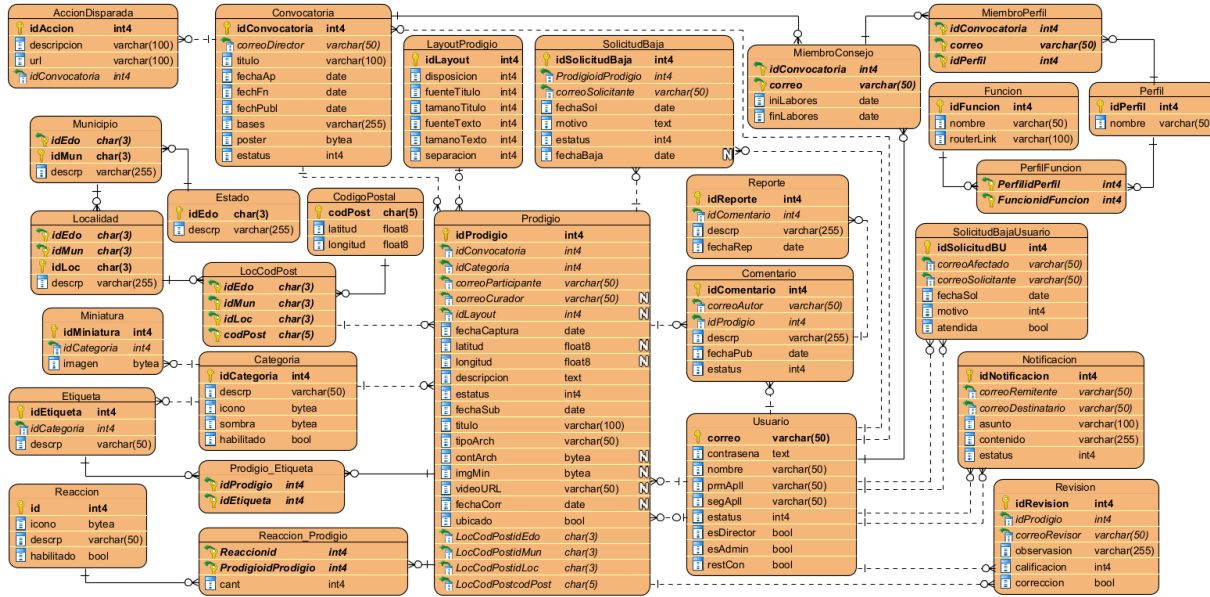


Figura 3.8 Diagrama entidad relación de la base de datos

Es importante señalar que algunos campos de la tabla de prodigios se encuentran marcados como *Nulleable*, esto es requerido así porque dichos campos se llenarán conforme los procesos de revisión y curaduría avancen y los miembros del consejo realicen sus tareas.

Posteriormente, con las tablas ya definidas, se inició con el diseño de los procesos almacenados que, mayormente, se encargan de realizar conversiones de archivos de binario a cadena base64 y viceversa, así como de realizar transacciones en las que se involucra la inserción de múltiples registros en la base de datos al mismo tiempo, pues de haber un error u ocurrir la interrupción del proceso, es necesario que la base de datos regrese al estado anterior de la ejecución del proceso. Cabe señalar que, si bien Django es capaz de soportar transacciones en la base de datos a partir de su implementación mapeo - objeto - relacional (ORM por sus siglas en inglés), se decidió delegar el trabajo de realizar transacciones en la base de datos al gestor para aligerar la

carga de trabajo a la aplicación. Con esto en mente, los procesos establecidos fueron los siguientes:

- **Filtro de Galería:** Búsqueda en los registros de prodigios bajo parámetros específicos como el tipo de archivo, título, etiquetas asociadas, convocatoria. La función regresa los campos de idprodigio, título y miniatura (codificado como cadena base64) de un listado de los prodigios que coincidan con los parámetros introducidos.
- **Convocatorias Recientes:** Regresa de la base los campos de idconvocatoria, título y póster (codificado en base64) de datos un listado de las convocatorias que se encuentre dentro de los 3 meses más recientes.
- **Detalle Prodigio:** Regresa los campos de un prodigio, especificado por su campo identificador, incluyendo los que representan archivos almacenados como cadenas base 64.
- **Autorregistro:** Crea un nuevo registro de usuario en la base de datos empleando la función MD5 para codificar la contraseña indicada por el usuario.
- **Iniciar Sesión:** Identifica la existencia de un usuario dado dentro de la tabla Usuario y auth_user, esta última generada por parte del sistema de autenticación de *Django REST Framework*. Una vez encontrado, obtiene los roles que ostenta y, de existir, elimina el token anterior al que se encontrara asociado el usuario en la tabla *authuser_token*. Finalmente, retorna la clave del usuario junto a una serie de banderas que identifican los permisos del usuario dentro de la plataforma.
- **Registrar Prodigio:** Genera un nuevo registro de prodigio con la información proporcionada por el usuario. Cabe señalar que la información que representa los archivos se recibe en forma de cadena base64 y se decodifica a binario para almacenarlo en la base de datos. La excepción la constituyen los videos, cuyo contenido no se almacena en la base de datos sino en Google Drive, en este caso se guarda en la base el identificador generado por dicho almacenamiento (además del resto de los campos propios del prodigio).

- **Corregir Prodigio:** Actualiza los campos de un prodigio, específicamente el título, descripción y categoría asociada, con base en la información proporcionada por el usuario.
- **Detalle Convocatoria:** Retorna el campo de la tabla convocatoria que almacena el contenido del archivo del póster de la convocatoria.
- **Registrar Convocatoria:** Genera un nuevo registro de convocatoria en la base de datos con la información proporcionada por el usuario. El campo referente a la imagen del póster de la convocatoria se recibe como cadena base64 y se codifica a binario para almacenarse en la base de datos.
- **Registrar Categoría:** Permite crear en la base de datos un nuevo registro de categoría junto con sus miniaturas relacionadas. Las miniaturas se reciben como un arreglo de cadenas base64 y se insertan tras registrar la categoría.
- **Detalle Categoría:** Regresa los campos de una categoría especificada por su identificador, codificando los campos de la tabla que almacenan archivos como cadenas base64, así como un listado de las miniaturas asociadas a la categoría. Al igual que en el caso de la categoría, los campos que almacenan archivos son codificados como cadenas base64.
- **Actualizar Categoría:** Actualiza los campos de una categoría especificada por su identificador, así como las miniaturas asociadas a esta. Los campos que almacenan archivos se reciben como cadena base64 y las miniaturas se reciben como arreglo de cadenas base64. Internamente, el proceso actualiza los campos de la categoría para posteriormente eliminar todas las miniaturas asociadas y reinsertar las nuevas como miniaturas asociadas a la categoría.
- **Reestablecer Contraseña:** Actualiza un registro de usuario cambiando su contraseña actual por una nueva. Retorna el registro de usuario actualizado.
- **Listar por Convocatoria:** Lista todos los prodigios pertenecientes a una convocatoria especificada por su identificador, junto con un conteo de todas las revisiones con las que cuenta cada prodigio.

- **Actualizar Etiquetas:** Actualiza el listado de etiquetas de un determinado prodigio, especificado por su identificador. Internamente, elimina todas las etiquetas asociadas al prodigio y reinserta un nuevo conjunto de etiquetas.
- **Registrar o Actualizar Perfil:** Recibe los parámetros necesarios para ingresar un nuevo tipo de Perfil de Usuario en la base de datos, o bien, actualizar uno ya existente. Dado que recibe entre sus parámetros una serie de funciones a las que el perfil recibido tendrá acceso, la transacción también almacena las relaciones entre el perfil y sus funciones asociadas, y en el caso de que sea una actualización, elimina primero las funciones asociadas anteriores.
- **Filtrar Convocatoria:** Función que realiza búsquedas de convocatorias con base en parámetros específicos dependiendo del tipo de usuario que emplee la función. Si es Revisor, busca aquéllas en las que participa como Revisor; si es Curador, busca aquéllas en las que participa como Revisor; si es Director, busca aquéllas registradas por el usuario; y si se trata de Público, busca todas las convocatorias que se encuentren con la Galería Abierta.
- **Usuarios por Filtro:** Realiza una búsqueda de usuarios con base en su correo, nombre, primer apellido y/o segundo apellido.
- **Reaccionar Prodigios:** Registra una nueva reacción sobre un prodigio indicado. Internamente, si no existen reacciones asociadas al prodigio, crea un nuevo registro de la reacción seleccionada asociada al prodigio indicado con la cantidad iniciada en 1. Si el prodigio ya posee una relación con la reacción seleccionada, aumenta en uno la cantidad.
- **Registrar Miembro:** Permite asignar a un usuario, previamente registrado, el o los roles seleccionados por el director.
- **Registrar Usuario Miembro:** Registra un nuevo usuario en la base de datos y le asigna el o los roles seleccionados por el director.
- **Evaluación Prodigio:** Obtiene la lista de prodigios de una convocatoria específica con estatus junto con la cantidad de votos a favor, en contra y solicitudes de corrección que posee.

- **Filtrar Prodigio:** Busca los prodigios que coincidan con una serie de parámetros proporcionados por el usuario como la categoría asociada, el tipo de archivo del prodigio y la localización (por municipio y localidad).
- **Cantidad de Comentarios por Fecha:** Obtiene la cantidad de comentarios realizados dentro de un rango de fechas establecido, agrupados por semanas. Si en una o más semanas específicas no se realizaron comentarios, la cantidad obtenida es 0.
- **Prodigios más Comentados:** Busca los 3 prodigios más comentados, así como la cantidad de comentarios de cada uno, en un rango de fechas definido.
- **Cantidad de Reacciones:** Regresa la cantidad de veces que cada reacción registrada se realizó sobre algún prodigio.
- **Prodigios más Reaccionados:** Obtiene los 3 prodigios más reaccionados y la cantidad de reacciones, de cada reacción registrada, realizadas sobre cada prodigio identificado.
- **Prodigios Recientes:** Busca el id, título e imagen de los 5 prodigios más recientes de la base de datos. El campo que almacena la imagen se codifica en cadena base64 para ser regresada como texto.

3.2.4 Sprint 4. Codificación de los servicios de Django Rest Framework

En esta vuelta se realizó la creación del entorno de trabajo de Django REST Framework para iniciar con la codificación del servidor. Para esto, se creó un nuevo directorio bajo el nombre de “server” dentro del cual, a través de la terminal de comandos, se creó un nuevo entorno virtual de Python, en su versión 3.11, por medio del comando **py -m venv env**. El resultado de esta acción es un nuevo directorio llamado “env” en el que se encuentran las bibliotecas básicas de Python, en *sprints* posteriores se incluyeron otras.

Posteriormente, se activó el entorno virtual con el comando **env\Scripts\activate** (encontrándose posicionado en la carpeta raíz del proyecto). Tras esto, se instalaron las bibliotecas necesarias para el funcionamiento del servidor. A continuación, se describen brevemente:

- **Django (pip install django)**

Marco de trabajo para el desarrollo de aplicaciones rápido, en el lenguaje de programación Python.

- **Django rest framework (pip install djangorestframework)**

Marco de trabajo para Django que ofrece diversas herramientas para el desarrollo de API Web. Se emplea para facilitar las tareas de codificación de vistas, serialización de datos y autenticación del API.

- **psycopg2 (pip install psycopg2-binary)**

Adaptador del gestor de base de datos PostgreSQL más usado de Python, diseñado para aplicaciones con muchos subprocesos que crean y destruyen múltiples cursores o realizan grandes cantidades de operaciones INSERT o DELETE de forma simultánea.

- **PyDrive (pip install PyDrive)**

Biblioteca que contiene, simplifica y encapsula en clases las tareas de google-api-python-client de la API de Google Drive, como la autenticación de OAuth2.0. En este proyecto se emplea para cargar y consultar videos, con previa autenticación, a una cuenta de Google drive para su posterior consulta.

- **django-cors-headers (pip install django-cors-headers)**

Biblioteca que agrega a las respuestas del servidor cabeceras de Intercambio de Recursos de Origen Cruzado (CORS por sus siglas en inglés) para permitir solicitudes en el visualizador web a la aplicación Django desde orígenes distintos. Se instaló para permitir las solicitudes entre las aplicaciones de Angular y Django, debido a que, en desarrollo, se encuentran en servidores diferentes. Para la versión final en el despliegue, se permite únicamente el acceso desde el mismo servidor (en este caso, en el hospedaje de *Digital Ocean*).

Una vez instaladas las bibliotecas básicas, se crearon tanto el proyecto como la aplicación de *Django REST Framework* a desarrollar. Para esto, se ejecutó, en primer lugar, el comando **django-admin startproject prodigios**, lo que generó un nuevo directorio llamado “prodigios”. Seguidamente, se ingresó al archivo creado y se ejecutó el comando **django-admin startapp**

prodigios_app. La estructura de directorios del proyecto hasta este punto se muestra en la Figura 3.9.

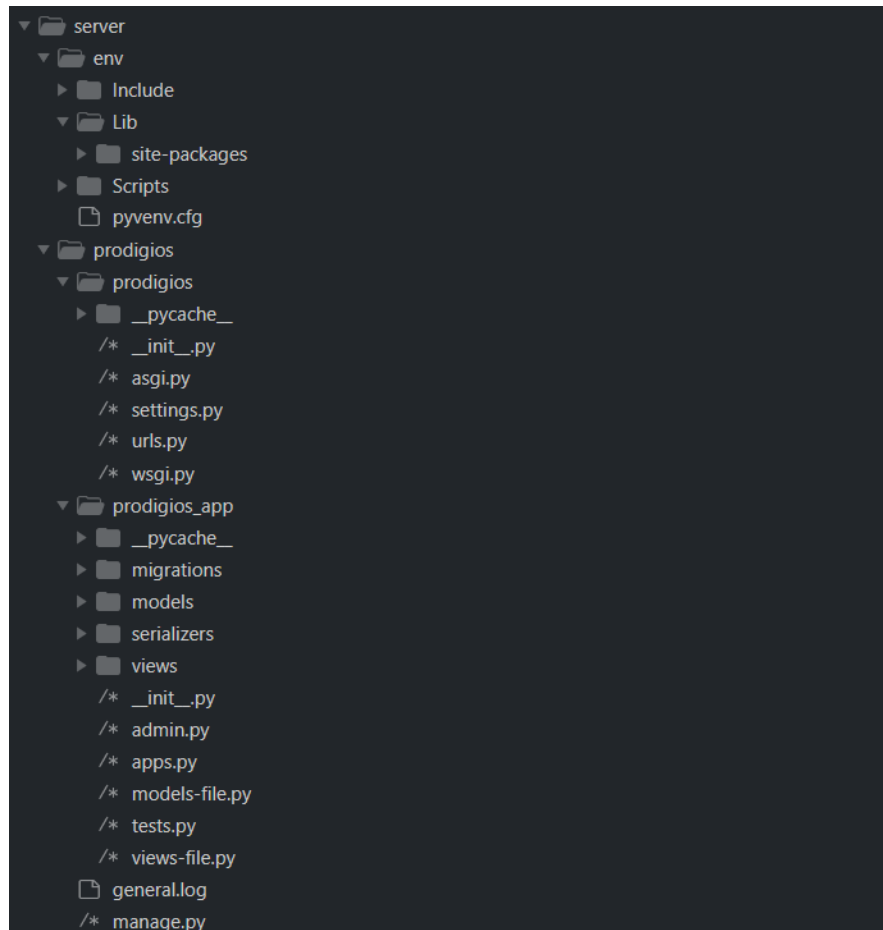


Figura 3.9 Estructura de los archivos del proyecto (servidor)

Posteriormente, se inició con la codificación de las clases del sistema, que en el contexto de Python se les conocen como modelos. Para agilizar el proceso de codificación de los modelos se empleó el comando **python manage.py inspectdb > models.py** para obtenerlos con base en las tablas de la base de datos y posteriormente corregirlos en función del diseño planteado anteriormente. El resultado de esta acción es un archivo de Python llamado **models.py** en el que se encuentra el código de todos los modelos generados.

Dada la extensa cantidad de modelos necesarios para la aplicación, se decidió separar el contenido del archivo **models.py** en múltiples archivos dentro de un paquete. Los paquetes de

la aplicación se implementaron como un directorio en el que se creó el archivo `__init__.py`. Los archivos de los modelos se encuentran dentro de su directorio correspondiente y son referenciados dentro del archivo `__init__.py` para emplearlos en otros recursos haciendo importaciones desde el paquete y no haciendo referencia a la dirección física de los archivos. Cabe mencionar que la nomenclatura para los nombres de los archivos es completamente en minúsculas, separando palabras por medio de un “_”, como se muestra en la Figuras 3.10 y 3.11.

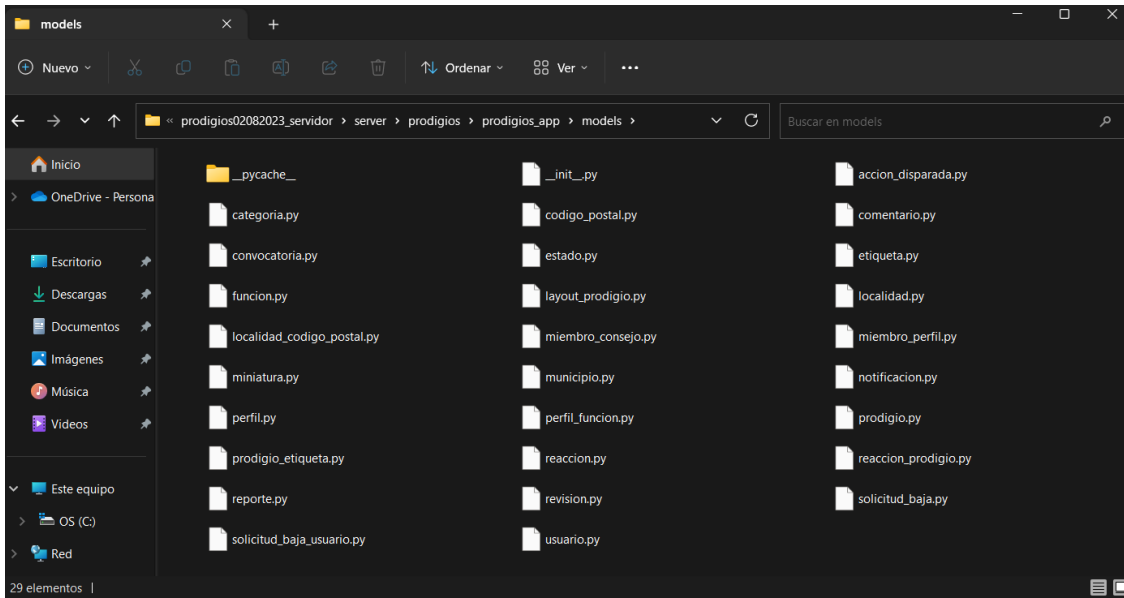


Figura 3.10 Contenido de la carpeta "modelos"

```

__init__.py
1  from prodigios_app.models.accion_disparada import AccionDisparada
2  from prodigios_app.models.categoria import Categoria
3  from prodigios_app.models.codigo_postal import CodigoPostal
4  from prodigios_app.models.comentario import Comentario
5  from prodigios_app.models.convocatoria import Convocatoria
6  from prodigios_app.models.estado import Estado
7  from prodigios_app.models.etiqueta import Etiqueta
8  from prodigios_app.models.funcion import Funcion
9  from prodigios_app.models.layout_prodigio import LayoutProdigio
10 from prodigios_app.models.localidad import Localidad
11 from prodigios_app.models.localidad_codigo_postal import LocCodPost
12 from prodigios_app.models.miembro_consejo import MiembroConsejo
13 from prodigios_app.models.miembro_perfil import MiembroPerfil
14 from prodigios_app.models.miniatura import Miniatura
15 from prodigios_app.models.municipio import Municipio
16 from prodigios_app.models.notificacion import Notificacion
17 from prodigios_app.models.perfil import Perfil
18 from prodigios_app.models.perfil_funcion import PerfilFuncion
19 from prodigios_app.models.prodigio import Prodigio
20 from prodigios_app.models.prodigio_etiqueta import ProdigioEtiqueta
21 from prodigios_app.models.reaccion import Reaccion
22 from prodigios_app.models.reaccion_prodigio import ReaccionProdigio
23 from prodigios_app.models.reporte import Reporte
24 from prodigios_app.models.revision import Revision
25 from prodigios_app.models.solicitud_baja import SolicitudBaja
26 from prodigios_app.models.solicitud_baja_usuario import SolicitudBajaUsuario
27 from prodigios_app.models.usuario import Usuario

```

Figura 3.11 Contenido del archivo `__init__.py`

Internamente, cada modelo es una subclase de la clase **django.db.models.Model** de Python, mientras que cada uno de sus atributos representa un campo de la base de datos (Listado 3.1), por lo que es posible indicar cuáles de estos corresponden a la clave primaria, y los límites de campos según su tipo (longitud en el caso de cadenas, valor máximo y mínimo en el caso de números).

Listado 3.1 Contenido interno de una clase del proyecto

```

1  from django.db import models
2
3  class Categoria(models.Model):
4      idcategoria = models.AutoField(primary_key=True)
5      descrip = models.CharField(max_length=50)
6      icono = models.BinaryField()
7      sombra = models.BinaryField()
8      iconofa = models.CharField(max_length=50)
9      habilitado = models.BooleanField()
10     iconofa = models.CharField(max_length=50)
11
12     class Meta:
13         managed = False
14         db_table = 'categoria'
```

Tras la obtención del archivo models.py, se notó que en algunos modelos tenían errores al momento de generarse debido a la incapacidad de Django para trabajar con claves compuestas. Debido a esto, se realizaron algunos cambios en la base de datos para la implementación del ORM de acuerdo a las directrices de Django, por ejemplo, la eliminación de claves compuestas en las tablas de relación, sustituyéndolas por una restricción de unicidad entre las 2 claves foráneas que la conforma, y la inclusión de una clave primaria auto incremental. En el caso de las tablas que no representan exclusivamente relaciones entre 2 tablas, como las tablas de Municipio, Localidad y Código Postal en las que la clave primaria es compuesta para referenciar la relación entre municipios con estado, localidades con municipio y municipio con código postal, se decidió eliminar la composición de su clave primaria en pro de una clave auto incremental, dando como resultado el diagrama mostrado en la figura 3.12.

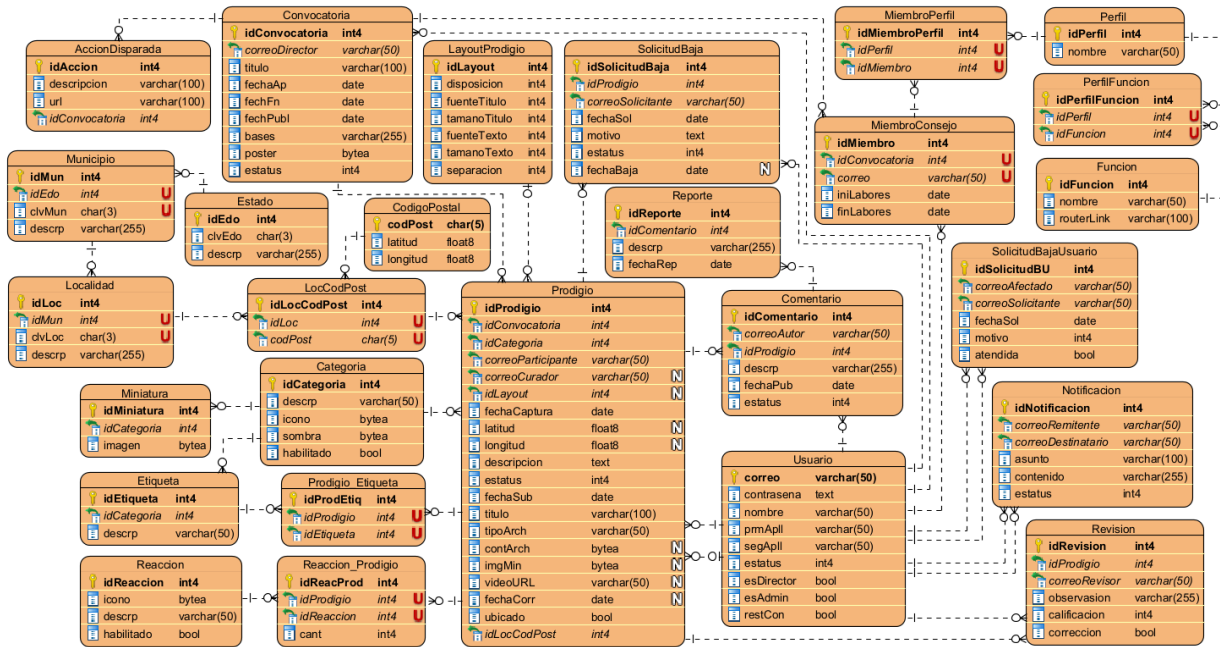


Figura 3.12 Modelo Actualizado de la Base de Datos

Una vez realizados los ajustes en la base de datos y los modelos correspondientes, se inició con la codificación de las vistas que exponen la funcionalidad a los servicios. Para esto, se creó un nuevo paquete llamado *views* en el que se almacenarán las vistas del servidor. Cabe aclarar que las vistas, pese a su nombre, no son representaciones visuales de la funcionalidad (pantallas), sino que se tratan de interfaces de servicios; el nombre que asigna Django es debido a su origen de marco basado en Modelo-Vista-Controlador. Cada vista creada hace referencia a un recurso, generalmente una clase diseñada del sistema, y es una subclase de la clase `rest_framework.viewsets.ViewSet`. En las vistas se definen las funciones que representarán los servicios expuestos del servidor (*endpoints*), y cada función devuelve un estatus HTTP de acuerdo a los resultados de su ejecución., pudiendo ser cualquiera de los códigos 200 (Todo está bien), 201 (Creado), 404 (No se encontró el recurso solicitado) o 500 (Hubo un error en el servidor y la solicitud no pudo ser completada). Como estándar de codificación, se decidió seguir la guía de estilo PEP 8, para Python [53]. En Listado 3.2 se muestra el código de una de las vistas generadas.

Listado 3.2 Código de una de las vistas de la aplicación

```

1  from prodigios_app.models import Prodigio
2  from prodigios_app.models import ProdigioEtiqueta
3  from prodigios_app.models import Etiqueta
4  from prodigios_app.serializers import EtiquetaSerializer
5  from prodigios_app.serializers import EtiquetaParamsSerializer
6  from prodigios_app.models import Categoria
7
8  class EtiquetaViewSet(viewsets.ViewSet):
9
10     logger = logging.getLogger(__name__)
11
12     """
13     """
14
15     @action(detail=False, methods=['get'],
16             url_path=r'por_prodigio/(?P<idProd>.+)' )
17     def listar_por_prodigio(self, request, pk=None, idProd=None): ...
18
19     """
20     """
21
22     @action(detail=False, methods=['get'],
23             url_path=r'por_categoria/(?P<idCat>.+)' )
24     def listar_por_categoria(self, request, pk=None, idCat=None): ...
25
26     """
27     """
28
29     @action(detail=False, methods=['post'], url_path=r'registrar',
30             permission_classes=[IsAuthenticated])
31     def registrar_etiqueta(self, request, pk=None): ...
32
33     """
34     """
35
36     @action(detail=False, methods=['patch'],
37             url_path=r'actualizar',
38             permission_classes=[IsAuthenticated])
39     def actualizar_etiqueta(self, request, pk=None): ...
40
41     """
42     """
43
44     def retrieve(self, request, pk=None): ...

```

Para las operaciones de búsqueda, inserción, actualización o eliminación de registros, de primera opción, se emplean los modelos mostrados anteriormente. Estos, al heredar de la clase

django.db.models.Model poseen, como funciones heredadas, las funciones *filter*, *get*, *create*, *update_or_create* y *delete* a través del atributo *objects* (Listado 3.3).

Listado 3.3 Ejemplo de inserción empleando los métodos heredados del modelo

```

1  @action(detail=False, methods=['post'], url_path=r'registrar',
2      permission_classes=[IsAuthenticated])
3  def registrar_etiqueta(self, request, pk=None):
4      try:
5          params = EtiquetaParamsSerializer(data={
6              'idetiqueta': 0,
7              'idcategoria': request.data['idcategoria'],
8              'descrip': request.data['descrip']
9          })
10         params.is_valid(raise_exception=True)
11         categoria = Categoria.objects.get(
12             idcategoria=request.data['idcategoria'])
13         etiqueta = Etiqueta.objects.create(
14             idcategoria=categoria,
15             descrip=params.data['descrip'])
16         retorno = EtiquetaSerializer(data={
17             'idetiqueta': etiqueta.idetiqueta,
18             'idcategoria': etiqueta.idcategoria.idcategoria,
19             'descrip': etiqueta.descrip
20         })
21         retorno.is_valid(raise_exception=True)
22         return Response(
23             retorno.data,
24             status=status.HTTP_201_CREATED)
25     except Error as e:
26         return Response({
27             'clase': 'EtiquetaViewSet',
28             'funcion': 'registrar_etiqueta',
29             'error': str(e.__cause__)},
30             status=status.HTTP_500_INTERNAL_SERVER_ERROR)

```

Cuando se da el caso de operaciones complejas sobre la base de datos, como la conversión de cadenas base64 o la realización de transacciones en las que se realizan múltiples operaciones de inserción o eliminación, en lugar de los modelos de Django se invoca a las funciones almacenadas descritas en el punto 3.2.3 de este capítulo. Para incorporar estos recursos en la aplicación de Django se emplea la biblioteca *django.db*, para ejecutar código SQL desde Django, como se muestra en el Listado 3.4, de las líneas 4 a 8.

Listado 3.4 Incorporación de proceso almacenados en la aplicación

```

1  def retrieve(self, request, pk=None):
2      try:
3          if Convocatoria.objects.filter(idconvocatoria=pk).exists():
4              with connection.cursor() as cursor:
5                  cursor.execute(
6                      """SELECT * FROM detalle_convocatoria(%s)""" ,
7                      [pk])
8                  convocatoriaRow = cursor.fetchone()
9                  oConvocatoria = Convocatoria.objects.get(idconvocatoria=pk)
10                 director = UsuarioSerializer(data={
11                     'correo': oConvocatoria.correodirector.correo,
12                     'nombre': oConvocatoria.correodirector.nombre,
13                     'prmapll': oConvocatoria.correodirector.prmapll,
14                     'segapll': oConvocatoria.correodirector.segapll})
15                 director.is_valid(raise_exception=False)
16                 convocatoria = ConvocatoriaSerializer(data={
17                     'idconvocatoria': oConvocatoria.idconvocatoria,
18                     'titulo': oConvocatoria.titulo,
19                     'fechaap': oConvocatoria.fechaap,
20                     'fechfn': oConvocatoria.fechfn,
21                     'fechpubl': oConvocatoria.fechpubl,
22                     'bases': oConvocatoria.bases,
23                     'poster': convocatoriaRow[1],
24                     'estatus': oConvocatoria.estatus,
25                     'correodirector': director.data})
26                 convocatoria.is_valid(raise_exception=False)
27                 return Response(convocatoria.data, status=status.HTTP_200_OK)
28             else:
29                 return Response(status=status.HTTP_404_NOT_FOUND)
30         except Error as e:
31             self.logger.error(str(e.__cause__))
32             return Response({
33                 'clase': 'ConvocatoriaViewSet',
34                 'funcion': 'retrieve',
35                 'error': str(e.__cause__)},
36                 status=status.HTTP_500_INTERNAL_SERVER_ERROR)

```

Como se menciona en la sección 3.2.2, dado que la conversión de vídeos a cadenas base64 resulta poco viable debido a los tiempos, se opta por subir los archivos en formato de vídeo a una cuenta de *Google Drive*. Para lograr esto, en primer lugar, se requiere el uso del API de *Google Drive*, para lo cual es necesario obtener las credenciales necesarias de OAuth 2.0 de la consola de *Google API* [54].

Una vez obtenidas las credenciales, en formato *json*, es necesario crear el archivo de configuración *setting.yaml* para permitir que *pydrive* guarde las credenciales de OAuth de forma permanente como se muestra en el Listado 3.5.

Listado 3.5 Contenido del archivo de *settings.yaml*.

```

1  client_config_backend: settings
2  client_config:
3    client_id: [omitido]
4    client_secret: [omitido]
5
6  save_credentials: True
7  save_credentials_backend: file
8  save_credentials_file: credential_module.json
9
10 get_refresh_token: True
11
12 oauth_scope:
13   - https://www.googleapis.com/auth/drive

```

Tras esto, deben ejecutarse las líneas de código mostradas en el Listado 3.6, que se encuentran en el archivo *quickstar.py*, para que se genere el módulo de credenciales para *pydrive*. Esta ejecución se aprecia en la Figura 3.13. Tras esto, se proporcionará un enlace a una pantalla de verificación de usuario para validar que la autenticidad de la cuenta propietaria de las credenciales OAuth, como se muestra en la Figura 3.14.

Listado 3.6 Código para la generación del módulo de credenciales.

```

1  from pydrive.auth import GoogleAuth
2
3  gauth = GoogleAuth()
4  gauth.LocalWebserverAuth()

```

```
(env) C:\Users\adolf\Documents\Mestría\prodigios02082023_servidor\prodigios02082023_servidor\server\prodigios\prodigios_app\drive>py quickstart.py

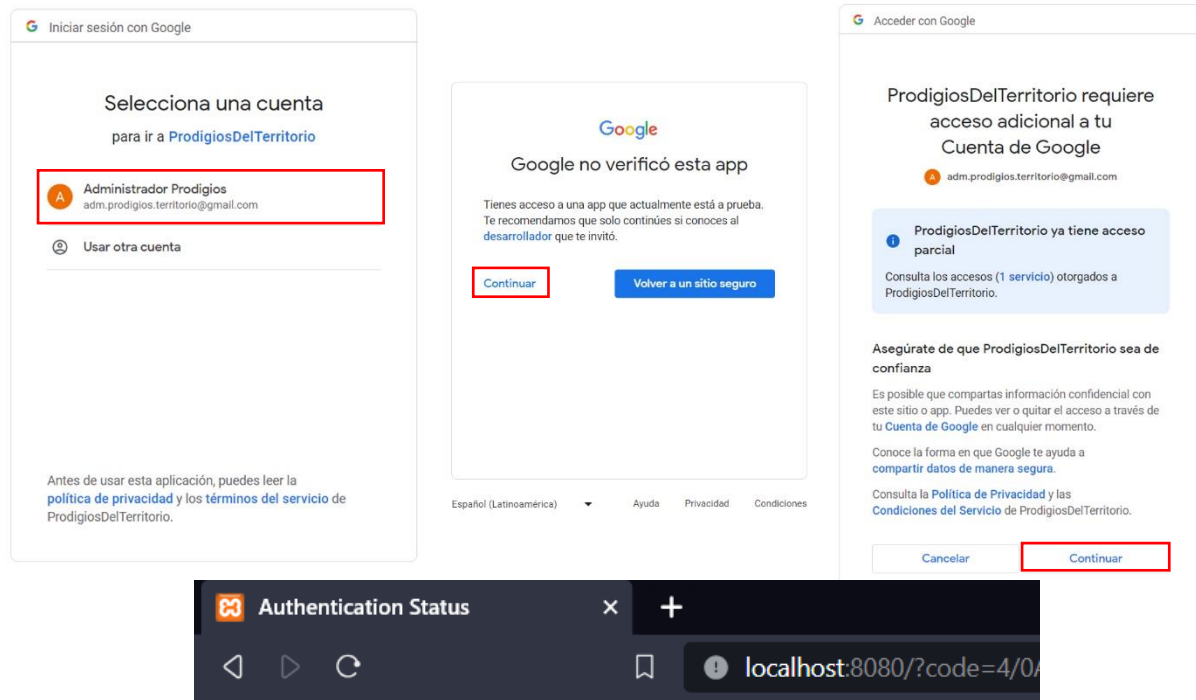
(env) C:\Users\adolf\Documents\Mestría\prodigios02082023_servidor\prodigios02082023_servidor\server\prodigios\prodigios_app\drive>py quickstart.py
C:\Users\adolf\Documents\Tests\server\env\Lib\site-packages\oauth2client\helpers.py:255: UserWarning: Cannot access credential_module.json: No such file or directory
  warnings.warn(_MISSING_FILE_MESSAGE.format(filename))
Your browser has been opened to visit:

  https://accounts.google.com/o/oauth2/auth?client_id=564564550446-52r80c020tpctp29h6lagug6b2947uu1.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&access_type=offline&response_type=code&approval_prompt=force

Authentication successful.

(env) C:\Users\adolf\Documents\Mestría\prodigios02082023_servidor\prodigios02082023_servidor\server\prodigios\prodigios_app\drive>
```

Figura 3.13 Ejecución del archivo quickstart.py



The authentication flow has completed.

Figura 3.14 Pantalla de autenticación para generar credenciales del API de Google Drive

Para subir los vídeos a *Google Drive* se emplea un módulo creado específicamente para proveer de las funciones de inicio de sesión, con las credenciales generadas, y carga de archivos a *Google Drive*. Este último se muestra en el Listado3.7.

Listado 3.7 Función para la subida de archivos de vídeo.

```

1  def uploadFile(temporalName, newName):
2      idVideo = ''
3      credenciales = login()
4      archivo = credenciales.CreateFile({
5          'parents': [{
6              "kind": "drive#fileLink",
7              "'Id de la carpeta de drive a la que se subirá'"
8              "id": "[omitido]"}]})
9      "'Nombrar el archivo que se subirá'"
10     archivo['title'] = '+newName
11     "'Busca el archivo subido en el servidor'"
12     archivo.SetContentFile('drive/temporal/'+temporalName)
13     archivo.Upload()
14     "'Obtén el id del archivo, generado por drive,
15     por su nombre'"
16     idVideo = search_id("title = '"+newName+"'")
17     return idVideo

```

3.2.5 Sprint 5. Codificación de la aplicación Angular

En esta iteración se inició con la codificación de la aplicación del lado del cliente. Como se mencionó en secciones anteriores, se emplea el marco de trabajo de Angular. Para empezar, se instaló el comando npm, versión 9.1.3, en el sistema y posteriormente se instalaron las versiones 18.12.1 y 15.1.5 de *Node* y Angular respectivamente. Para agilizar el proceso de codificación en la aplicación cliente, se decidió emplear la plantilla gratuita sakai-ng [55], basada en la biblioteca de componentes de interfaz de usuario *primeNG*, ofrecida en la página oficial de Angular. Una vez descargada la plantilla, se ejecutó el comando **npm install** para instalar las dependencias requeridas para el correcto funcionamiento de la plantilla, así como la biblioteca *leaflet* para el uso de mapas dentro de la plataforma.

3.2.5.1 Estructura de los archivos del proyecto

La plantilla de sakai-ng propone una aplicación separada en dos módulos principales: uno para los elementos como el menú lateral, superior y pie de página, representado por la carpeta *layout*, y otro para los elementos de código que brindan la funcionalidad de la aplicación, en este caso representado por la carpeta *platform*. Dentro de la carpeta *platform* se encuentran las carpetas

api, donde se almacenan las clases que se emplean en la aplicación; *components*, en la que se encuentran los componentes empleados por la aplicación; misc, abreviación de miscelánea, en donde se alojan elementos de código con funciones diversas empleadas por varios componentes; y *services*, en la cual se alojan los archivos requeridos para el funcionamiento de los servicios web que la aplicación emplea para conectarse con el API de *Django REST Framework*, como se observa en la Figura 3.15. En Angular, un servicio hace referencia a una clase cuya función es compartir datos o lógica de negocio con otros componentes, esto último gracias al uso del decorador *@Injectable*, lo que permite inyectar dicha funcionalidad en otros componentes o módulos de la aplicación.

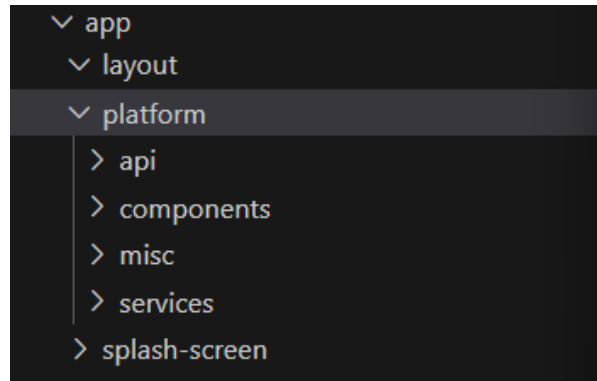


Figura 3.15 Estructura de la carpeta *platform*.

Siguiendo con la estructura de archivos propuesta por la plantilla, en la carpeta *components*, los componentes de la aplicación Angular se organizan en módulos separados que representan a cada tipo de usuario. En cada uno de estos módulos se crea, también, un módulo de enrutamiento (*routing module*) que permite complementar las rutas a los componentes internos del módulo, ejemplo de esto se aprecia en la Figura 3.16. Los componentes de cada módulo representan la funcionalidad de los casos de uso con los que cada tipo de usuario interactúa en la plataforma, y para los casos en los que múltiples tipos de usuario acceden al mismo caso de uso, se emplea el módulo *platform-common*. Adicionalmente, y derivado de diversas consideraciones identificadas durante el desarrollo del cliente, se crearon módulos para las pantallas de preguntas frecuentes y una pantalla *splash* que se muestra cada vez que se carga la aplicación.

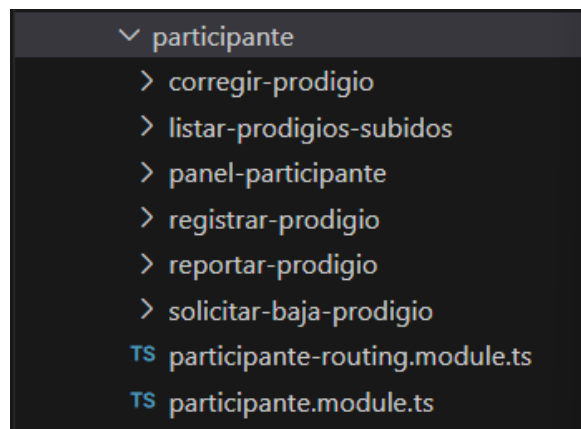


Figura 3.16 Estructura del módulo de Usuario Participante.

En cada componente, representado por una carpeta, se encuentran una plantilla html y un archivo de lógica *typescript*. Adicionalmente, cada componente cuenta con un módulo de enrutamiento para poder ser incluidos en la ruta base de la aplicación. Un ejemplo de la estructura de los componentes se aprecia en la Figura 3.17.

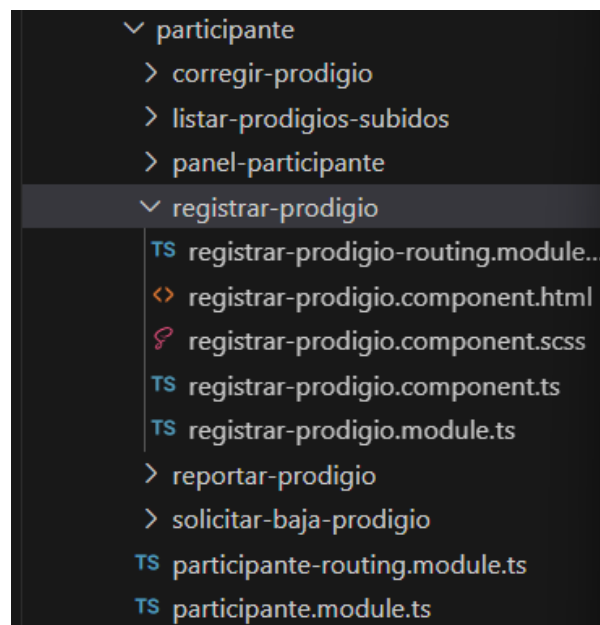


Figura 3.17 Estructura del Componente Registrar Prodigio.

Para acceder a los *endpoints* del API de Django, se generaron los servicios necesarios dentro de la carpeta *service*. Para cada recurso del servidor se emplea un archivo de servicio nombrado haciendo referencia al recurso al que accederá el servicio seguido de la palabra “*service*” y

finalizando con la extensión de archivo de *typescript* (ts), separados por un punto, por ejemplo: *categoria.service.ts*. Estos se alojan en la carpeta *service*, mencionada al inicio del punto 3.2.5.1.

3.2.5.2 Contenido de los elementos de código de la aplicación

La base de la plataforma se encuentra en su capacidad de enviar y recibir información para cumplir con los pasos de la metodología de Prodigios del territorio. Es por esto que primero se desarrollaron los servicios que se comunican con el API creada en el punto 3.2.4. Para realizar solicitudes al servidor desde la aplicación Angular, específicamente desde los servicios, se emplea la case *HttpClient* [56], disponible como clase inyectable con los métodos necesarios para realizar solicitudes HTTP (*get*, *post*, *put*, *patch* y *delete*). Para utilizarla dentro del servicio, se recibe como inyección de dependencias a través de un parámetro tipo *HttpClient* en el constructor de la clase del servicio con el modificador de acceso *private* como se observa en el Listado 3.8, en la línea 13.

Listado 3.8 Importación y Uso de la clase *HttpClient*.

```

1  import { Injectable } from '@angular/core';
2  import { HttpClient, ... } from "@angular/common/http";
3  import { Observable, throwError } from 'rxjs';
4  import { catchError, map } from 'rxjs/operators';
5  import { SERVER_URL } from '../misc/constants';
6  ...
7
8  @Injectable({
9    providedIn: 'root'
10 })
11 export class ProdigiosService {
12
13   constructor(private http: HttpClient) { }
14
15   ...
16
17 }
```

Para realizar las solicitudes HTTP, se invoca la función adecuada por medio del parámetro definido en el constructor. Cada función requiere de una ruta a la cual realizar la solicitud, en este caso la dirección base se representa por medio de la constante *SERVER_URL*, definida en

el archivo “constantes” ubicado en la carpeta misc (Figura 3.18), junto con un conjunto de objetos distintos que puede o no recibir.

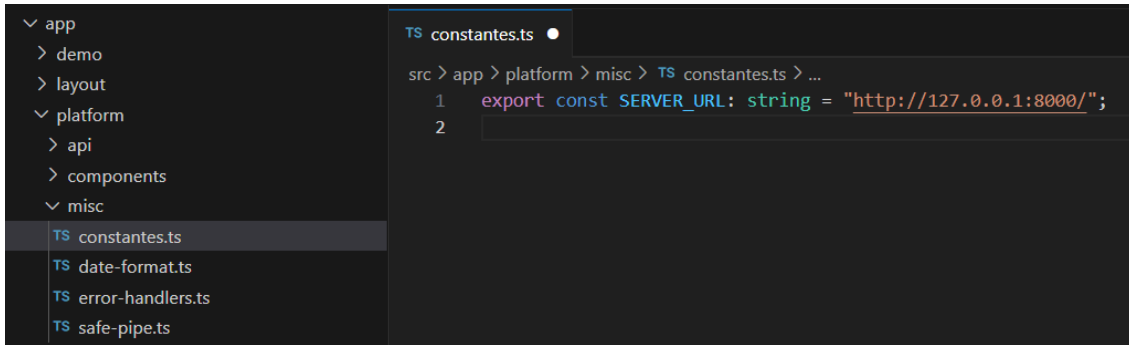


Figura 3.18 Contenido del archivo constantes

Por ejemplo, en la función *get* en esta aplicación en particular, además de la ruta del API, es posible anexar a la solicitud un objeto *HttpParams* para definir la lista de parámetros que el *endpoint* requiere para su funcionamiento, así como un objeto *HttpHeaders* que especifique las cabeceras empleadas para realizar las solicitudes al API. Para este proyecto se emplea la cabecera *Authentication* para enviar al API un *Token* que permite validar la sesión del usuario actualmente activo. Existen solicitudes en las que no es requerido dicho *Token*, sin embargo, en caso de requerirlo y no enviarlo, o bien, enviar uno cuya validez haya caducado para el API, se recibirá como respuesta un error del servidor, evitando así accesos no autorizados a recursos que se plantean solo visibles para usuarios registrados y con sesiones activas. Ejemplos del uso de la función *get* se pueden observar en el Listado 3.9, en las líneas 1, de la línea 8 a la 23 y de la línea 25 a la 40.

Como anotación adicional, todas las funciones en las que se realiza una solicitud HTTP son definidas por el tipo *Observable*, una implementación de la biblioteca RxJS para manejar peticiones y respuestas AJAX, pues todos los métodos de la clase *HttpClient* retornan un *observable*.

Listado 3.9 Distintas formas del uso de la función *get*.

```

1  public verDetalleProdigio(idProdigio:number): Observable<any> {
2      return this.http.get(SERVER_URL+"prodigios/"+idProdigio)
3      .pipe(
4          catchError(ErrorHandlers.handleHttpError)
5      )
6  }
7
8  public listarProdigiosParaGaleria(params:GaleriaProdigioParams) {
9      let queryParams = new HttpParams();
10     queryParams = queryParams.append("tipoarch", params.tipoarch);
11     queryParams = queryParams.append("etiquetas", params.etiquetas);
12     queryParams = queryParams.append("titulo", params.titulo);
13     queryParams = queryParams.append("categoria", ''+params.categoria);
14     queryParams = queryParams.append("convocatoria",
15         ''+params.convocatoria);
16
17     return this.http.get(SERVER_URL+"prodigios/galeria",
18         {
19             params: queryParams
20         }).pipe(
21             catchError(ErrorHandlers.handleHttpError)
22         )
23 }
24
25 public listarProdigiosSubidos(correoUsuario:string): Observable<any> {
26     let sesion:Sesion = JSON.parse(
27         sessionStorage.getItem('Sesion') as string
28     ) as Sesion;
29     const httpOptions = {
30         headers: new HttpHeaders({
31             Authorization: 'Token '+sesion.token
32         })
33     };
34     return this.http.get(
35         SERVER_URL+"prodigios/por_usuario/"+correoUsuario+"/",
36         httpOptions
37     ).pipe(
38         catchError(ErrorHandlers.handleHttpError)
39     )
40 }

```

Para el caso de las funciones *post* y *patch*, al igual que *get*, reciben como parámetro obligatorio la ruta del servidor, teniendo como parámetros adicionales el objeto *HttpHeaders* para representar las cabeceras de la solicitud, así como un objeto del tipo *FormData* que agrupa los campos del formulario de datos requerido para los métodos HTTP mencionados. Ejemplo de esto se pueden apreciar en el Listado 3.10 y 3.11.

Listado 3.10 Ejemplo del uso de la función post.

```

1 public registrarProdigio(prodigio:ProdigioParams): Observable<any> {
2   let sesion:Sesion = JSON.parse(
3     sessionStorage.getItem('Sesion') as string
4   ) as Sesion;
5   const httpOptions = {
6     headers: new HttpHeaders({
7       Authorization: 'Token '+sesion.token
8     })
9   };
10  const formData = new FormData();
11  formData.append("titulo", prodigio.titulo);
12  formData.append("idconvocatoria", ''+prodigio.idconvocatoria);
13  formData.append("idcategoria", ''+prodigio.idcategoria);
14  formData.append("correoparticipante", prodigio.correoparticipante);
15  formData.append("fechacaptura", prodigio.fechacaptura);
16  formData.append("tipoarch", prodigio.tipoarch);
17  formData.append("contarch", prodigio.contarch);
18  formData.append("imgmin", prodigio.imgmin);
19  formData.append("ubicado", ''+prodigio.ubicado);
20  formData.append("latitud", prodigio.latitud);
21  formData.append("longitud", prodigio.longitud);
22  formData.append("idloccodpost", ''+prodigio.idloccodpost);
23  formData.append("descripcion", ''+prodigio.descripcion);
24  formData.append("video", prodigio.video);
25
26  return this.http.post<any>(
27    SERVER_URL+'prodigios/autenticado/', formData, httpOptions
28  ).pipe(
29    catchError(ErrorHandlers.handleHttpError)
30  )
31 }

```

Listado 3.11 Ejemplo del uso de la función patch.

```

1 public corregirProdigio(prodigio:Prodigio): Observable<any> {
2   let sesion:Sesion = JSON.parse(
3     sessionStorage.getItem('Sesion') as string) as Sesion;
4   const httpOptions = {
5     headers: new HttpHeaders({
6       Authorization: 'Token '+sesion.token
7     })
8   };
9   const formData = new FormData();
10  formData.append("idprodigio", ''+prodigio.idprodigio);
11  formData.append("idcategoria",
12    ''+prodigio.idcategoria.idcategoria);
13  formData.append("descripcion", prodigio.descripcion);
14  formData.append("titulo", prodigio.titulo);
15
16  return this.http.patch<any>(
17    SERVER_URL+'prodigios/correccion/autenticado/',
18    formData,
19    httpOptions
20  ).pipe(
21    catchError(ErrorHandlers.handleHttpError)
22  )
23 }

```

Con respecto al contenido de la carpeta *components*, como se explicó al inicio de esa sección, esta incluye los módulos que representan los tipos de usuario disponibles en la aplicación. En los módulos de cada tipo de usuario se importan los respectivos archivos de enrutamiento, de modo que los componentes de cada módulo sean reconocidos por el módulo principal. Por su parte, en el módulo de enrutamiento de cada tipo de usuario se definen las rutas que redirigen a los componentes a los que el respectivo tipo tiene acceso, de modo que la ruta completa se conforme por el fragmento de ruta del módulo principal, seguida del fragmento del módulo del usuario como se aprecia en los Listados 3.12 y 3.13.

Listado 3.12 Ejemplo de Ruta definida en el módulo de enrutamiento principal.

```

1  import { RouterModule } from '@angular/router';
2  import { NgModule } from '@angular/core';
3  ...
4  @NgModule({
5    imports: [
6      RouterModule.forRoot(
7        [ { path: '', component: AppLayoutComponent,
8          children: [
9            ...
10           { path: 'participante', loadChildren: () => import(
11             './platform/components/participante/participante.module'
12             ).then(m => m.ParticipanteModule) },
13           ...
14         ]
15       }, ... ],
16     ], ...
17   )
18 ],
19 exports: [RouterModule]
20 })
21 export class AppRoutingModule { }

```

Listado 3.13 Ejemplo de Ruta definida en el módulo de participante.

```

1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3
4  const routes: Routes = [
5    {
6      path: 'panel_participante',
7      data: { breadcrumb: 'Panel Participante' },
8      loadChildren: () => import(
9        './panel-participante/panel-participante.module'
10     ).then(m => m.PanelParticipanteModule)
11   },
12   ...
13 ];
14
15 @NgModule({
16   imports: [RouterModule.forChild(routes)],
17   exports: [RouterModule]
18 })
19 export class ParticipanteRoutingModule { }

```

Cabe señalar que, para reconocer y utilizar parámetros dentro de un componente, es necesario especificar el nombre con el que se referirá a los parámetros recibidos anteponiendo el símbolo “:” en el módulo de enrutamiento, específicamente, en el atributo *path* del arreglo de objetos *Routes* que define el fragmento de ruta al que está asociado el componente. Los parámetros se obtienen de la ruta escrita en la barra de direcciones del visualizador web. Ejemplo de esto se observa en el Listado 3.14.

Listado 3.14 Contenido del módulo de enrutamiento del componente Corregir Prodigio.

```

1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { CorregirProdigioComponent }
4    from './corregir-prodigio.component';
5
6  const routes: Routes = [
7    {
8      path: ':idProdigio', component: CorregirProdigioComponent
9    }
10 ];
11
12 @NgModule({
13   imports: [RouterModule.forChild(routes)],
14   exports: [RouterModule]
15 })
16 export class CorregirProdigioRoutingModule { }
```

En el módulo de cada componente, como en el caso de los módulos de usuarios, se referencia su respectivo módulo de enrutamiento en el atributo *imports* y, además, los componentes de *primeNG*, biblioteca sobre la que se diseñó la plantilla base del proyecto, *sakai-ng*, mencionada al inicio del punto 3.2.5, que se ocupen en el respectivo componente pues es necesario para ser reconocidos como etiquetas válidas en su plantilla html. Por ejemplo, como se muestra en la Listado 3.15, en el componente Subir Prodigio se utilizan campos para captura de campos de fecha, para los que se emplea el módulo de calendarios de *primeNG* (*CalendarModule*); campos de texto, para los que se importan los módulos *InputTextModule* e *InputTextareaModule*; campos para la carga de archivos, para los que se importa el módulo *FileUploadModule*; entre otros. Además, en el atributo *declarations*, se referencia el archivo *typescript* del componente.

Listado 3.15 Módulo de enrutamiento del componente Subir Prodigio.

```

1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { RegistrarProdigioRoutingModule }
4    from './registrar-prodigio-routing.module';
5  import { RegistrarProdigioComponent }
6    from './registrar-prodigio.component';
7  import { CalendarModule } from 'primeng/calendar';
8  import { InputTextareaModule } from 'primeng/inputtextarea';
9  import { InputTextModule } from 'primeng/inputtext';
10 import { FileUploadModule } from 'primeng/fileupload';
11
12 @NgModule({
13   declarations: [RegistrarProdigioComponent],
14   imports: [
15     CommonModule,
16     RegistrarProdigioRoutingModule,
17     CalendarModule,
18     InputTextareaModule,
19     InputTextModule,
20     FormsModule,
21     FileUploadModule,
22   ]
23 })
24 export class RegistrarProdigioModule { }

```

Por parte de los archivos de lógica de los componentes, en cada uno de ellos se importan los servicios que el componente emplea para su funcionamiento. En el caso del componente Corregir Prodigio, importa los servicios de Prodigios, Revisiones y Categorías, pues los emplea para buscar el prodigio seleccionado, así como subir los cambios realizados sobre este, obtener la lista de observaciones realizadas por el equipo de revisores y la lista de categorías disponibles para el formulario de corrección respectivamente. Adicionalmente, se importan las clases *ActivatedRoute* y *Router* para monitorear los parámetros obtenidos de la ruta escrita en la barra de navegación del visualizador web, en el caso de *ActivatedRoute*, y redirigir la navegación de un componente a otro, en el caso de *Router*.

Dado que tanto los servicios y las clases de navegación son inyectables, estos se definen como parámetros en el constructor con el modificador de acceso *private*, como el caso de la clase

HttpClient mencionado anteriormente. Para acceder a los parámetros de la ruta se emplea el atributo *snapshots.params* de la clase *ActivatedRoute*, especificando dentro de corchetes el nombre del parámetro definido en el módulo de enrutamiento del componente como se muestra en el Listado 3.16, en las líneas 17 y 18.

Listado 3.16 Importaciones y su implementación en el componente *Corregir Prodigio*.

```

1  import { Component, ViewChild } from '@angular/core';
2  import { ActivatedRoute, Router } from '@angular/router';
3  ...
4
5  @Component({
6    selector: 'app-corregir-prodigio',
7    templateUrl: './corregir-prodigio.component.html',
8    styleUrls: ['./corregir-prodigio.component.scss'],
9    providers: [MessageService]
10 })
11 export class CorregirProdigioComponent {
12   ...
13   constructor(
14     private router: Router,
15     private activatedroute: ActivatedRoute,
16     ...) {
17     this.prodigio.idprodigio =
18       this.activatedroute.snapshot.params["idProdigio"];

```

En cuanto a la utilización de los servicios dentro de los componentes, estos se llaman a través del nombre de parámetro definido en el constructor para dicho Servicio. Cada función que se suscribe a un servicio tiene dos posibles resultados derivados de la ejecución del servicio: la primera, la respuesta exitosa del servidor y, por ende, la continuación del flujo normal del caso de uso; y la segunda, la obtención de un código de error como respuesta del API, lo que deriva en un flujo alternativo o excepción del caso de uso.

Por ejemplo, la función “listarCategorías”, del componente *Corregir Prodigio*, espera obtener un arreglo de categorías del API que se mostrará en un elemento de lista desplegable del formulario de corrección de prodigios. Para esto, dentro de esta función se suscribe a la función “listarCategorias” del servicio de categorías. Si el servicio se ejecuta en su totalidad, asigna la respuesta obtenida a la variable que almacena el arreglo de categorías presentado en el

formulario y, en caso de recibir un código de error, despliega un mensaje de alerta que notifique del error al solicitar el servicio y redirige el visualizador web a la pantalla de listado de convocatorias. Ejemplo de esto se aprecia en el Listado 3.17.

Listado 3.17 Ejemplo del uso de servicios en la función listarCategorias.

```

1  listarCategorias() {
2    this.categoriasService.listarCategorias().subscribe({
3      next:(resp: any) => {
4        this.categorias = resp;
5        this.categoriaSeleccionada =
6          this.categorias.filter(
7            (categoria) =>
8              categoria.idcategoria ===
9                this.prodigio.idcategoria.idcategoria
10           )[0];
11        this.searching = false;
12      },
13      error:(error) => {
14        alert('No se pudo obtener la lista de Categorias');
15        this.router.navigate(['/common/listar_convocatorias']);
16      }
17    });
18  }

```

Para presentar los elementos multimedia en los componentes se emplearon las etiquetas tradicionales de imagen y audio, junto con la etiqueta *iframe* para presentar tanto documentos pdf como vídeo. Para insertar las cadenas base 64 como recurso (*src*) de la etiqueta, se especifica el tipo y extensión del archivo a mostrar junto con el tipo de codificación de la cadena que se está usando como recurso, por ejemplo, si se desea mostrar una imagen codificada en una cadena base64, en el atributo *src* de la etiqueta *img* de HTML se indica de la siguiente forma: **data:image/jpg;base64, /9j/4AAQSkZJRgABAQEA...** Esta forma de presentar en las plantillas HTML funciona tanto para imágenes, audio y documentos pdf, sin embargo, para el caso de los vídeos, la forma de presentar es un tanto distinta. Como ejemplo para la presentación de los documentos, audio e imagen se muestra el Listado 3.18, a partir de las líneas 11, 19 y 30 respectivamente.

Listado 3.18 *Presentación de multimedia en la plataforma.*

```

1  <div
2    *ngIf="this.prodigio.tipoarch=='Documento/PDF'"
3    style="position:relative; width:100%; margin-top:-1.2rem">
4    <div
5      style="width:100%;
6        background: #fdfeff;
7        height: 3.5rem;
8        position:absolute;
9        z-index:1">
10   </div>
11   <iframe frameborder="0"
12     type="application/pdf"
13     [src]='data:application/pdf;base64, '
14       + this.prodigio.contarch | safe"
15     style="width:100%; height:30rem; border:0;" >
16   </iframe>
17 </div>
18
19 <audio
20   *ngIf="this.prodigio.tipoarch=='Audio/MP3'"
21   controls
22   controlsList="nodownload"
23   [src]="cadenaAudio">
24 </audio>
25
26 <div style="display: flex;
27   justify-content: center;
28   align-items: center;
29   height: 100%;">
30   <img
31     *ngIf="this.prodigio.tipoarch=='Imagen/JPG'"
32     [src]='data:image/png;base64, '
33       + this.prodigio.contarch"
34     alt="Imagen"
35     style="max-width: 100%; max-height: 100%;"/>
36 </div>

```

Como se mencionó en la sección 3.2.2, la conversión de video a cadena base64 toma mucho más tiempo que otros tipos de archivo por lo que se optó subirlos al almacenamiento en la nube por medio de una cuenta de *Google Drive*. Como resultado de esta acción, se obtiene el

identificador con el que la nube identifica al vídeo que se guarda en el registro del prodigio asociado y es por medio de este que se accede al material desde la aplicación. Para presentar los videos se emplea el tag *iframe* pasando como recurso el enlace del vídeo embebido generado por *Google Drive*, de la forma: [https://drive.google.com/file/d/\[idVideo\]/preview](https://drive.google.com/file/d/[idVideo]/preview), en donde el [idVideo] se sustituye por el valor asociado al prodigio.

Cabe mencionar que cuando se intenta emplear enlaces embebidos, como es el caso del enlace generado para los videos, es posible que salte un error debido a la inspección de seguridad de la aplicación Angular. Para solucionar este problema es necesario crear un *Pipe*[57] que sustituya la configuración de los metadatos de una clase e importarse en el módulo del componente donde sea necesario utilizar. En este caso el *Pipe* se encarga de convertir un enlace inseguro a seguro. Todas las *Pipes* necesitan implementar la clase *PipeTransform* e inyectar la clase *Domsanitizer*[58], que es la que se encarga de prevenir errores de seguridad de secuencias de comandos entre sitios por medio de la función *bypassSecurityTrustResourceUrl*. Finalmente, en el decorador *Pipe*, se asigna un nombre para poder utilizarse en el DOM. El código del *Pipe* y su implementación en el DOM se observan en el Listado 3.19 y 3.20 respectivamente.

Listado 3.19 *Pipe codificada para la sanitización de enlaces embebidos.*

```

1  import { Pipe, PipeTransform } from '@angular/core';
2  import { DomSanitizer } from '@angular/platform-browser';
3
4  @Pipe({
5    name: 'safe'
6  })
7  export class SafePipe implements PipeTransform {
8    constructor(private sanitizer: DomSanitizer) { }
9
10   transform(url:any) {
11     return this.sanitizer.bypassSecurityTrustResourceUrl(
12       url
13     );
14   }
15 }
```

Listado 3.20 *Presentación de videos en la plataforma.*

```

1  <div
2    *ngIf="this.prodigio.tipoarch=='Video/MP4'"
3    style="position: relative;">
4    <iframe
5      [src]="url | safe" style="width:100%; height:25rem"
6      allow="autoplay" frameborder="0" scrolling="no"
7      seamless="">
8    </iframe>
9    <div
10     style="width: 80px; height: 80px; position: absolute;
11         opacity: 0; right: 0px; top: 0px;">
12     &nbsp;
13   </div>
14 </div>

```

3.2.6 Sprint 6. Pruebas de la ejecución del proceso de la plataforma

En este punto se refiere a pruebas integrales, puesto que cada elemento de cada capa se probó de manera individual. Para empezar, se realizaron pruebas sobre los componentes relacionados con la captura, revisión, evaluación y curación de prodigios de la plataforma. Posteriormente, se prueba la presentación del contenido en la plataforma. De lo más destacable a mencionar, son algunos detalles que se encontraron en diversas pantallas, como es el caso de algunos filtros de la galería que requería de todos los campos de búsqueda llenos para filtrar los registros, siendo que, en principio, tiene que ser posible filtrar aun si no se han capturado datos en el formulario de búsqueda, en cuyo caso se obtendrán todos los registros. Se corrigieron, además, algunos errores presentes en los carruseles de imágenes de las galerías, pues estos no rotaban las imágenes de forma natural, sino que siempre regresaban al elemento inicial cuando se seleccionaba la opción de recorrer las imágenes como se muestra en la Figura 3.19.

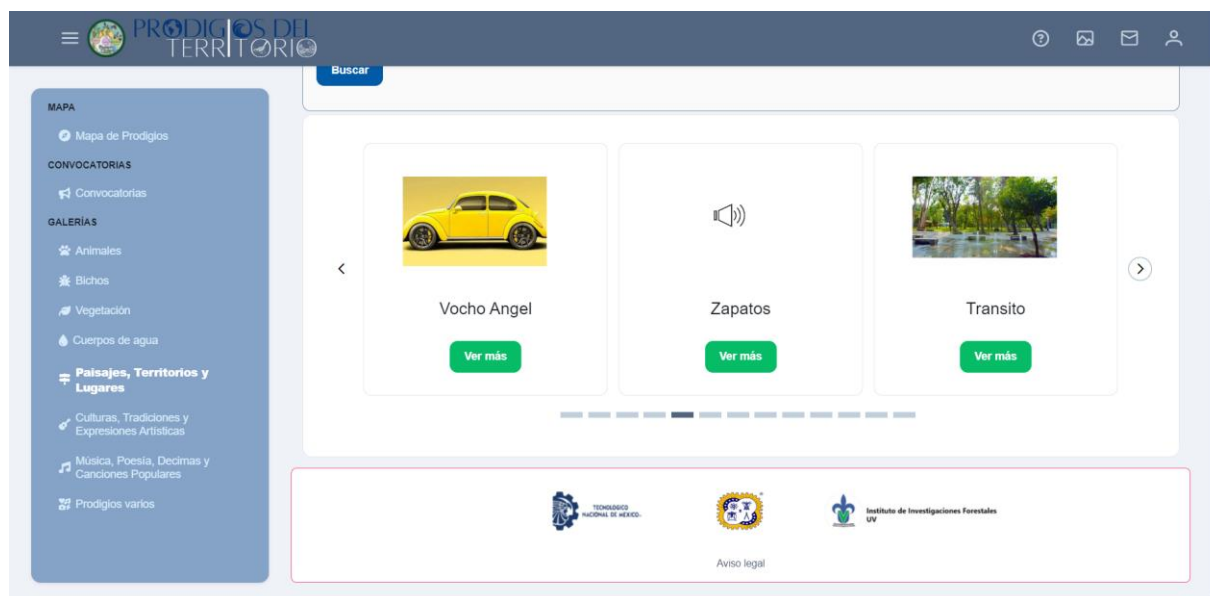


Figura 3.19 Carrusel de imágenes de la galería

Por otra parte, en cuanto al proceso principal de la plataforma, en la sección de Gestionar Consejo se encontró un pequeño error que ocurría al registrar un usuario nuevo como miembro del consejo (se esperaba incluir como miembro del Consejo tanto a usuarios existentes como a nuevos). Este error se dio debido a que, si bien el usuario se reflejaba en la tabla de usuarios diseñada para el sistema, este no se registra en la tabla que emplea el marco de trabajo de *Django REST Framework* para validar las sesiones de usuario.

En cuanto al proceso de curaduría, tras probar con la selección de fuentes inicial seleccionada para asignar la presentación de los prodigios, se observó que algunas de las fuentes no eran reconocidas por los visualizadores web, por lo que se optó por ajustar la selección con base en las fuentes más empleadas reconocidas por la mayoría de visualizadores web. Finalmente, en los listados de municipios, localidades y estados, al ser demasiado extensos se optó por permitir el filtrado de valores directamente en la lista con el fin de agilizar la selección de la localización de los prodigios. Este cambio aplicó tanto en los filtros de la galería como en el formulario de registro de prodigios y se observa en la figura 3.20.

Envía tu Prodigio

Título

Nombre

Fecha de Captura

Fecha de Captura

Categoría

Seleccione una Categoría

Tipo de Archivo del Prodigio

Seleccione un Tipo

Cargar Prodigio

+ Seleccionar X Cancelar

Ubicar en el Mapa

☒ Omitir Ubicación en el mapa

Estado

VERACRUZ

Municipio

Seleccione un Municipio

CASTILLO DE TEAYO

CATEMACO

CAZONES DE HERRERA

CERRO AZUL

CHACALTANGUIS

+ Seleccionar X Cancelar

Figura 3.20 Listado de municipios (previo a la corrección)

Como cierre de esta sección cabe señalar que, si bien el proceso principal de la plataforma se revisó de forma que estuviese lo más pulido posible para la realización de pruebas, aun quedaron pendientes la revisión de funciones, principalmente relacionadas a tareas del administrador, e imprevistos que se observaron al momento de evaluar el desempeño de la plataforma en producción, los cuales fueron atendidos en posteriores ciclos de trabajo.

Capítulo 4 Resultados

En este capítulo se presenta la implementación de la plataforma en un entorno real, con un grupo de usuarios de prueba, y los resultados obtenidos. El caso de estudio tuvo lugar en el taller “Encuentro Comunitario de Prodigios del Territorio”, en Jáltipan, Veracruz, realizado los días 9 y 10 de noviembre del 2023. El objetivo de esta implementación fue probar la usabilidad de la plataforma con un público conformado tanto por personas acostumbradas al uso de herramientas digitales como ajenos a éstas. Derivado de la implementación se obtuvieron diversas observaciones por parte de los participantes. A continuación se describe lo realizado.

4.1 Caso de Estudio: Encuentro Comunitario de Prodigios del Territorio

En esta sección se describen las actividades realizadas durante el caso de estudio realizado.

4.1.1 Planteamiento del caso de estudio

El caso de uso se llevó a cabo en el municipio de Jáltipan, Veracruz, en compañía de la directora del proyecto de tesis, la maestra Beatriz Alejandra Olivares Zepahua, así como la codirectora, la doctora Elsa Antonia Pérez Paredes. El acuerdo se realizó derivado de una serie de reuniones realizadas con un grupo de pobladores del municipio que, a partir del programa de televisión Veracruz Agropecuario, se interesaron en el proyecto y propusieron realizar el caso de estudio en el municipio para compartir los prodigios de su localidad.

4.1.2 Características del caso de estudio

La prueba de la plataforma contó con la asistencia de 23 participantes:

- 4 estudiantes del tecnológico de Minatitlán.
- 4 docentes del tecnológico de Minatitlán.
- 15 pobladores del municipio de Jáltipan.

El equipo de asistencia tecnológica se conformó por 4 miembros;

- La directora de tesis.
- El propio tesista.
- 3 residentes de Ingeniería en Sistemas Computacionales, asociados al proyecto.

Las herramientas tecnológicas que se emplearon para el caso de estudio consistieron de los siguientes elementos generales:

- 4 equipos de Cómputo con el sistema operativo Windows 11.

- Red Wifi con un ancho de banda de 5Mb.

4.1.3 Descripción de las actividades realizadas

En la primera parte del taller se realizaron algunas dinámicas que permitieron la socialización entre el público y, posteriormente, se hizo una presentación en la que se explicó el concepto de Prodigios del Territorio y la evolución de la metodología asociada a este concepto, desde sus primeras actividades hasta el Encuentro Comunitario de Prodigios del Territorio en Jáltipan. En la segunda parte, se explicó a grandes rasgos los motivos del desarrollo de la plataforma y se describió el proceso que implementa la plataforma para soportar la metodología. La figura 4.1 muestra la presentación del planteamiento la plataforma y la figura 4.2 muestra la exposición del proceso de la metodología dentro de la plataforma.



Figura 4.1 Presentación sobre el planteamiento de la plataforma.

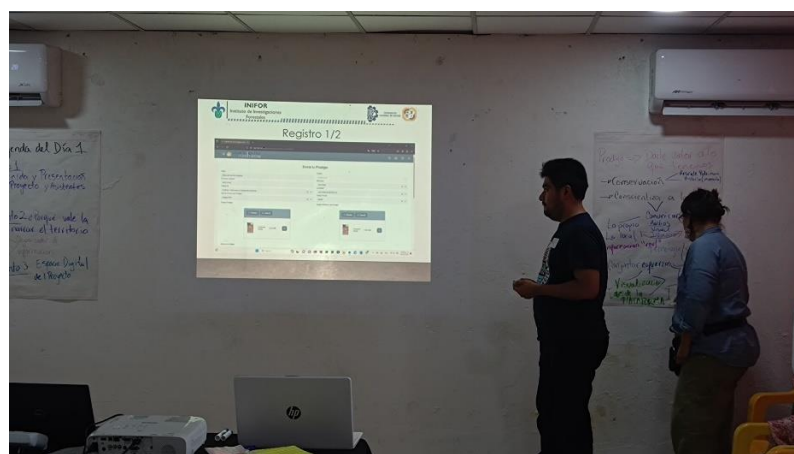


Figura 4.2 Presentación sobre el proceso de la plataforma.

Una vez finalizadas las presentaciones, se procedió a explicar algunas recomendaciones para el tratamiento de archivos multimedia previo a su incorporación en la plataforma. Se explicaron, entre otras cosas, cómo identificar qué extensión de archivo es posible subir (jpg, mp3, pdf y mp4), cuál es el tamaño de archivo permitido por cada tipo (500KB para imagen, 4MB para audio, 1MB para documento y 75MB para video) y cómo editar cada uno para ajustarlo a las limitaciones impuestas por la plataforma. Para cada tipo de archivo se recomendaron una serie de herramientas para llevar a cabo la tarea de edición multimedia sin la necesidad de descargar aplicaciones de terceros. Ejemplo de esta explicación se observa en la figura 4.3.



Figura 4.3 Explicación sobre la edición de imágenes.

En la siguiente fase, se invitó al público a participar en la publicación de prodigios de su territorio para el taller. Cada participante pasó con alguno de los asistentes, quienes los orientaron en la carga de prodigios a la plataforma y, en los casos que se necesitó, la edición de archivos. Posterior a este proceso, se agradeció a los participantes del taller y se concluyó con las actividades de ese día. Durante la tarde-noche el equipo de revisores y curadores, conformado por los miembros del equipo de asistencia tecnológica, se encargó de revisar, así como curar los prodigios recolectados para, finalmente, abrir la galería de la convocatoria generada para uso exclusivo del taller. Las actividades de asistencia por parte del equipo tecnológico se ilustran en las figuras 4.4 y 4.5.



Figura 4.4 Fotografía de parte del equipo de asistencia tecnológica.



Figura 4.5 Fotografías del equipo brindando orientación a los participantes.

Al iniciar el segundo día del taller, se mostró el resultado de las actividades realizadas el día anterior. Se presentaron en pantalla los prodigios compartidos por los participantes, ya visualizados en la galería y, además, se explicó en mayor profundidad el proceso general del funcionamiento de la plataforma, usando la propia plataforma, ya en línea, como apoyo visual. La descripción del proceso partió desde al alta de la convocatoria por parte del director; la carga de prodigios del participante; el proceso de revisión y corrección por parte de revisores y participantes; la curaduría de prodigios llevada por el equipo de curadores; finalizando con la apertura de la galería y la visualización del público. Las imágenes de esta actividad se aprecian en la figura 4.6.

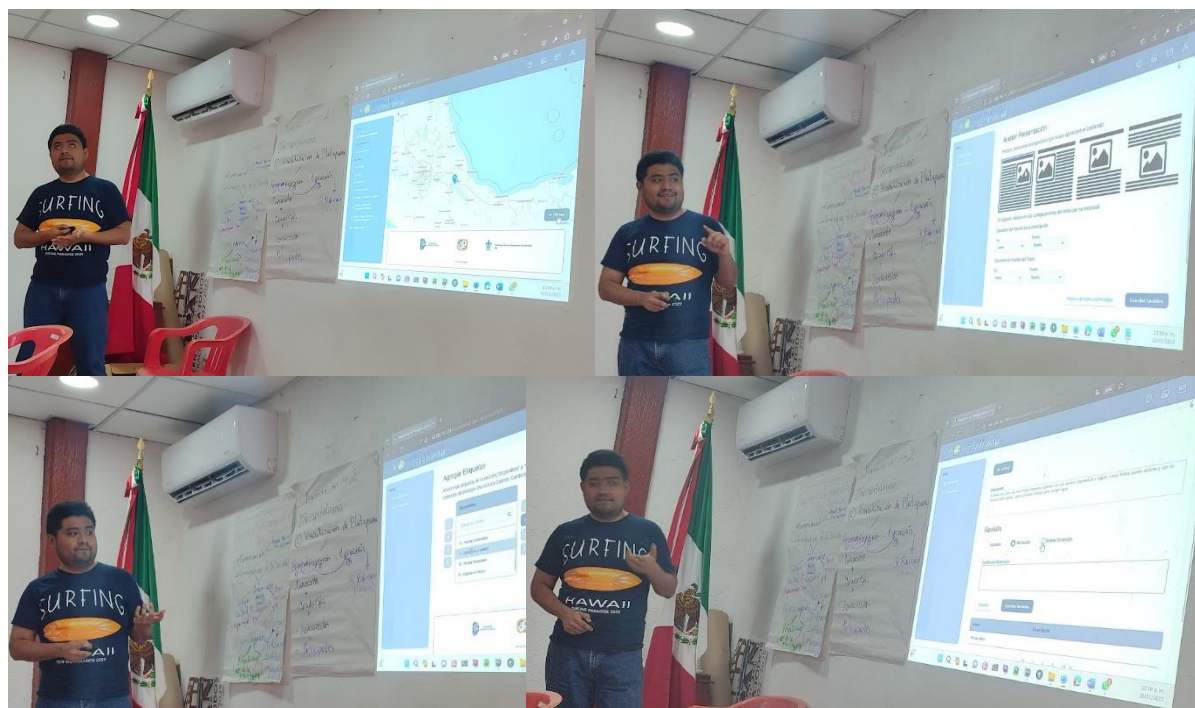


Figura 4.6 *Presentación de la plataforma en línea.*

Finalmente, se llevó a cabo una ronda de discusión sobre lo observado durante los dos días del taller y se alentó a los participantes a realizar de preguntas hacia el equipo tecnológico para resolver las dudas que aun persistían sobre el funcionamiento de la plataforma. Para cerrar, se solicitó al público que colaborara respondiendo de forma escrita una serie de preguntas con respecto a las actividades realizadas y, para cerrar, se solicitó al público que redactaran opiniones, observaciones y recomendaciones sobre el taller y la plataforma, esto a través de la invitación hacia los participantes para emplear la plataforma ahora que se encuentra disponible en línea.

4.1.4 Resultados obtenidos del caso de estudio

En líneas generales, la mayor parte de los participantes remarca en sus redacciones la importancia de proyectos como éste que fomentan el interés en mejorar, conservar y preservar la cultura, el entorno natural, las lenguas autóctonas, entre otros aspectos del territorio. Algunos otros participantes señalan que el tiempo de la presentación resultó insuficiente para explorar en profundidad los temas tratados durante las presentaciones, así como la explicación del funcionamiento de la plataforma.

En cuanto aspectos más específicos, durante el proceso de carga de prodigios por parte de los participantes, el equipo de asistencia técnica se encargó de apoyar a los usuarios en cualquier inquietud que el proceso descrito generara. En esta parte del taller, la mayor parte de las dudas presentadas por los participantes consistieron principalmente en cómo subir adecuadamente el material multimedia, sobre todo audio y vídeo, a la plataforma. “La plataforma está casi completa, pero falta la forma de editar prodigios, ya que hay personas que se les dificulta al momento de subir” comentó uno de los participantes de forma anónima en las recomendaciones redactadas.

Por otra parte, centrándose en el proceso seguido para registrar y gestionar las convocatorias en la plataforma, los principales problemas de diseño de pantallas se detectaron en los menús para la curaduría de prodigios, especialmente en la selección de la disposición en pantalla para el prodigio pues resulta poco intuitiva y explicativa sobre qué hacer en ella y cómo repercute en el prodigio que se está curando. Además, no quedaba del todo claro en plataforma cómo inscribirse para registrar convocatorias en la plataforma pues, para ser habilitado como director, es necesario que el administrador del sitio, explícitamente, haga el registro del usuario como director.

Como comentario final, es posible finalizar esta sección señalando que la principal causa de problemas identificados radica en la confusión que generan algunas pantallas debido a lo poco descriptivas que son en cuanto a la fase del proceso que pretenden sistematizar, considerando que el público en general desconoce la metodología Prodigios del Territorio. En el siguiente capítulo se abordan las recomendaciones consideradas para la resolución de estas observaciones y se cierra con las conclusiones finales del trabajo realizado en general.

Capítulo 5 Conclusiones y Recomendaciones

En este capítulo se detallan las conclusiones y recomendaciones obtenidas tras la ejecución del caso de estudio.

5.1 Conclusiones

Con base en lo analizado en el marco teórico y el estado del arte, es posible darse cuenta que pese a que existen antecedentes similares a la metodología de Prodigios del Territorio, resulta casi nula la existencia de procesos de naturaleza similar a la metodología como sistemas de software. Es posible considerar que esta situación es a causa de que la metodología posee una naturaleza de evolución constante, como se observó tras la identificación de los requerimientos funcionales, así como el diseño del sistema, lo que resulta en un desafío pues, dado que el proceso es propenso a recibir cambios relativamente frecuentes, resulta difícil realizar un diseño que considere y soporte dichos cambios constantes, lo que provoca una gran probabilidad de repetir el proceso de análisis y diseño para replantear la premisa del sistema.

En otros aspectos, tomando en cuenta los resultados obtenidos del caso de estudio, es posible resaltar, también, la diferencia en las necesidades de diseño de software destinado a uso empresarial y las de software de uso público enfocado a un propósito social, pues mientras que en el primero se da por hecho que el usuario está familiarizado, o como mínimo es capacitado, con el proceso que se busca sistematizar; con el segundo existe el desafío de considerar tanto a personas que no solo desconocen el proceso sistematizado, sino que también carecen de alguna experiencia o interacción con cualquier tipo de aplicación de software o dispositivo de cómputo. Esto es importante señalarlo, pues uno de los objetivos de la metodología es generar diálogo entre los distintos participantes que provienen de distintas formaciones y para esto es necesario que las interfaces de la plataforma sean lo más comprensibles y simples posibles con la finalidad de que sean usables tanto para personas con experiencia como ajenos al uso de aplicaciones web.

Centrándose en un aspecto más social, y derivado de las opiniones obtenidas por los participantes del caso de estudio, se puede apreciar la importancia que tiene para los habitantes de municipios como Jáltipan el compartir los aspectos de su territorio que consideran razón de

orgullo, especialmente considerando la problemática ambiental por la que pasan actualmente, relacionado con la calidad del agua que llega al municipio. Es importante recordar que uno de los objetivos de la metodología de Prodigios del Territorio es llevar, a futuro, a acciones que eventualmente generen un cambio en las formas en las que se reconoce, preserva y apropia el territorio, por lo que tener el interés y la motivación del público en el proyecto desde el lanzamiento es una buena señal, pues implica que existe un genuino interés en las personas por proyectos de resignificación del territorio.

Como conclusión, la plataforma mostró resultados prometedores gracias al interés mostrado por parte de los participantes del caso de estudio, tanto por el material presentado dentro de la plataforma, como por participar y tomar acción en las actividades que propone la misma. El programa, si bien presenta algunas situaciones en las cuales resulta poco intuitivo o poco explicativo con el proceso y las tareas que se realizan en ciertas pantallas, son cuestiones que se pueden arreglar agregando instrucciones más explícitas en dichas pantallas, así como hacer más visibles las secciones de preguntas frecuentes y manuales de uso en puntos específicos del proceso en los cuales surgieron la mayor cantidad de dudas. Como cierre, proyectos de esta índole poseen bastante potencial en la búsqueda de soluciones para mitigar la pérdida y el desinterés de la población general por su territorio pues, como se apreció en los resultados del caso de estudio, presentado de la manera correcta y con la instrucción adecuada, es posible incentivar al público a participar en actividades de reconocimiento del territorio y, posteriormente, dirigir ese interés a cuestiones de preservación y resignificación del mismo. El interés está allí, solo falta el espacio para compartir el conocimiento que se encuentra dispersado por todos los rincones del país.

5.2 Recomendaciones

Las recomendaciones específicas al programa se presentan a continuación:

- Módulo de estadísticas/desempeño de la plataforma: Si bien se planeó como una de las funciones de la plataforma, dado que el desarrollo y los esfuerzos se centraron más en el proceso principal de la metodología, quedó poco tiempo para desarrollarlo y asentarlo de forma final. Esta funcionalidad permitirá al administrador del sistema verificar el alcance que la plataforma logra, tanto en visualizaciones como participación del público, de forma

más sencilla y clara con el fin de obtener información de utilidad para tomar decisiones precisas sobre el rumbo de la metodología.

- Investigación más profunda sobre el diseño de las interfaces: Con el fin de ofrecer una mejor experiencia de usuario, se considera necesario investigar de forma más extensa el diseño de interfaces dirigidas a aplicaciones con propósitos sociales, no solo para facilitar la comprensión por parte de los usuarios, sino también para ser llamativo con la finalidad de atraer y retener la mayor cantidad de público nuevo.
- Pruebas con nuevos y más amplios grupos de trabajo: Si bien en el caso de estudio se obtuvieron buenos resultados que demostraron el interés del público por proyectos de naturaleza, es importante señalar que dicho grupo ya presentaba un interés previo por participar en el proyecto de Prodigios del Territorio. Aun es necesario probar la plataforma con grupos nuevos, más amplios y ajenos al proyecto de Prodigios del Territorio con el fin de verificar el interés y retención que es capaz de lograr el sistema por sí solo.

Anexos y Apéndices

Anexos

1. Anexo A: Carta de Usuario
2. Anexo B: Documento “Artefactos de Ingeniería de Software”
3. Anexo C: Documento “Diagrama de Actividades”
4. Anexo D: Documento “Narrativas de Casos de Uso”
5. Anexo E: Documento “Servicios de la Plataforma”

Anexo A: Carta de Usuario

Jáltipan de Morelos, Veracruz, a 13 de noviembre de 2023

Dr. Rafael Flores Peredo
Director del Instituto de Investigaciones Forestales (INIFOR) de la Universidad Veracruzana
PRESENTE

MAE Maricela Gallardo Córdova
Directora del Campus Orizaba del Tecnológico Nacional de México (ITO)
PRESENTE

Los C. Yasmín Galván Carreón y Eleazar Cortés Morales, representantes de Narradores y Narradoras de Jáltipan, así como el C. Marcos Morales Mistega y el Ing. Emiliano Pereira Hernández, representantes de la Comisión en Defensa del Pueblo de Jáltipan, les agradecemos el haber realizado en nuestro pueblo el "1er Encuentro Comunitario de Prodigios del Territorio", así como habernos capacitado en el uso de la plataforma digital "Prodigos del Territorio"; donde participaron, por parte del INIFOR, la Dra. Elsa Antonia Pérez Paredes, y por el ITO la MCE Beatriz Alejandra Olivares Zepahua así como el maestrante Adolfo Meza Romero y los residentes Jocelyn Flores López, Braulio David Hernández Palagot y Claudia Guadalupe Ramírez Lázares.

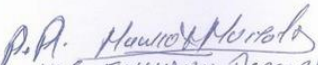
Con este trabajo las bellezas de nuestro municipio, Jáltipan, Veracruz, podrán ser conocidas en todas partes, así como también se podrán conocer los grandes problemas socioambientales. Sabemos que todavía falta mucho por hacer y que no hemos logrado una solución a estos problemas, pero confiamos en que dar a conocer la situación a mayor escala logrará que seamos escuchados.

ATENTAMENTE


Yasmín Galván Carreón
 Yasmín Galván Carreón
 Narradores y narradoras de Jáltipan, Veracruz


Eleazar Cortés Morales
 Eleazar Cortés Morales


MARCOS MORALES MISTEGA
 Marcos Morales Mistega
 Comisión en Defensa del Pueblo de Jáltipan, Veracruz


ING. EMILIANO PEREIRA HERNÁNDEZ
 Ing. Emiliano Pereira Hernández

Referencias

- [1] F. Ducarme y D. Couvet, “What does ‘nature’ mean?”, *Palgrave Communications*, vol. 6, núm. 1. Palgrave Macmillan Ltd., el 1 de diciembre de 2020. doi: 10.1057/s41599-020-0390-y.
- [2] R. Folch y J. Bru, “Ambiente, territorio y paisaje Valores y valoraciones”, 1946. [En línea]. Disponible en: www.editorialbarcino.cat
- [3] L. Llanos Hernández, “El concepto del territorio y la investigación en las ciencias sociales”, *Agricultura, Sociedad Y Desarrollo*, pp. 207–220, 2010.
- [4] M. Lascurain, T. Rivera-Núñez, C. A. González, y M. Medina García, “¿Qué significa biocultural?” Consultado: el 15 de agosto de 2022. [En línea]. Disponible en: <https://www.inecol.mx/inecol/index.php/es/ct-menu-item-25/ct-menu-item-27/17-ciencia-hoy/1570-que-significa-biocultural>
- [5] G. Estrada Hernández, “Sierra de Guadalupe, un espacio para todos. Curso-taller de formación de educadores ambientales (no formal)”, Universidad Autónoma de la Ciudad de México, Ciudad de México, 2021.
- [6] H. Mares, “Tipos de educación (Formal, no formal, informal)”, Academia. Consultado: el 15 de agosto de 2022. [En línea]. Disponible en: https://www.academia.edu/15362640/Tipos_de_educaci%C3%B3n_Formal_no_formal_informal_
- [7] M. Chacón-Ortiz, “El proceso de evaluación en educación no formal: Un camino para su construcción”, *Revista Electrónica Educare*, vol. 19, núm. 2, may 2015, doi: 10.15359/ree.19-2.2.
- [8] “Laboratorio Socioambiental Ciudadano”. Consultado: el 3 de septiembre de 2022. [En línea]. Disponible en: <https://labsocioambiental.com/>
- [9] Tecnológico de Monterrey, “EduTrends. Storytelling”, *Observatorio de Innovación Educativa del Tecnológico de Monterrey*. 2017.
- [10] P. Boullosa, *El corazón es un resorte: Metáforas y otras herramientas para mejorar nuestra educación*. Ciudad de México: Taurus, 2016.
- [11] Y. N. Harari, *Sapiens. De animales a dioses: Breve historia de la humanidad*. Barcelona: Debate, 2014.
- [12] “¿Qué es la escritura creativa? Características y aplicación en la actualidad”, UNIR - Universidad Internacional de La Rioja. Consultado: el 5 de septiembre de 2022. [En línea]. Disponible en: <https://www.unir.net/humanidades/revista/que-es-escritura-creativa/>

- [13] C. P. Elliot, “Principles for teaching creative writing at university”, *Arbor*, vol. 196, núm. 798, pp. 1–12, 2020, doi: 10.3989/arbor.2020.798n4002.
- [14] L. R. Dueñas Salmán y E. J. García López, “EL ESTUDIO DE LA CULTURA DE PARTICIPACIÓN, APROXIMACIÓN A LA DEMARCACIÓN DEL CONCEPTO”, *Razón y Palabra*, vol. 80, 2012, Consultado: el 3 de septiembre de 2022. [En línea]. Disponible en: <http://www.redalyc.org/articulo.oa?id=199524426008>
- [15] G. Aguilera Izaguirre y R. Blancarte Galván, “Análisis de la participación ciudadana en México”, *Ius Comitalis*, vol. 5, núm. 9, pp. 31–47, 2022, Consultado: el 3 de septiembre de 2022. [En línea]. Disponible en: <https://iuscomitalis.uaemex.mx/article/view/17007>
- [16] “Carta europea revisada sobre la participación de los jóvenes en la vida local y regional”, ene. 2017.
- [17] T. Vidal Moranta, “La apropiación del espacio: una propuesta teórica para comprender la vinculación entre las personas y los lugares”, *Anuario de Psicología*, vol. 36, pp. 281–297, 2005.
- [18] S. Al-Fedaghi, “Developing Web Applications”, *International Journal of Software Engineering and Its Applications*, vol. 5, núm. 2, pp. 57–68, 2011.
- [19] S. Luján-Mora, *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario. [En línea]. Disponible en: <https://www.researchgate.net/publication/320808713>
- [20] C. A. Ellis, S. J. Gibbs, y G. Rein, “Groupware: Some Issues and Experiences”, *Commun. ACM*, vol. 34, núm. 1, pp. 39–58, ene. 1991, doi: 10.1145/99977.99987.
- [21] M. S. Mikowski y J. C. Powell, *Single Page Web Applications: JavaScript End-to-End*, Primera Edición. Manning Publications Co., 2013.
- [22] E. Mougiakou *et al.*, “Participatory urban planning through online webGIS platform: Operations and tools”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, sep. 2020, pp. 831–834. doi: 10.1145/3428502.3428633.
- [23] V. Nisi, C. Silva, M. Dionísio, y N. J. Nunes, “A participatory platform supporting awareness and empathy building between tourists and locals: The Há-vita case study”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, sep. 2019. doi: 10.1145/3351995.3352049.
- [24] M. F. Morell, “How Much are Digital Platforms Based on Open Collaboration?: An analysis of technological and knowledge practices and their implications for the platform governance of a sample of 100 cases of collaborative digital platforms in

Barcelona”, en *Proceedings of the 14th International Symposium on Open Collaboration, OpenSym 2018*, New York, NY, USA: Association for Computing Machinery, Inc. 2018. doi: 10.1145/3233391.3233970.

- [25] M. Karhu, J. Häkkinen, y E. Timonen, “Collaborative media as a platform for community powered ecological sustainability transformations - A case study”, en *MUM 2019: 18th International Conference on Mobile and Ubiquitous Multimedia MUM 2019*), New York, NY, USA: Association for Computing Machinery, nov. 2019. doi: 10.1145/3365610.3368413.
- [26] R. Sánchez-Tortolero, J. Márquez-González, y G. Vérmaz-Hernández, “Participation and citizen empowerment platform for e-governance: Communal integration system (SINCO)”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, 2019, pp. 212–219. doi: 10.1145/3326365.3326393.
- [27] T. Cunningham Dahl-Jørgensen y E. Parmiggiani, “Platformization of the public sector: Assessing the space of possibility for participation”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, jun. 2020, pp. 35–39. doi: 10.1145/3384772.3385154.
- [28] A. Frid-Jimenez, J. Carson, A. Scott, P. Khantidhara, y D. Elza, “Designing Participedia: A Collaborative Research Platform”, en *Proceedings of the 16th Participatory Design Conference 2020-Participation(s) Otherwise - Vol. 2*, New York, NY, USA: Association for Computing Machinery, jun. 2020, pp. 21–25. doi: 10.1145/3384772.3385125.
- [29] B. Sanchez y M. X. Rodriguez-Paz, “Using BIM as a collaborative platform to improve e-learning in civil engineering”, en *2020 12th International Conference on Education Technology and Computers*, New York, NY, USA: ACM, oct. 2020, pp. 21–26. doi: 10.1145/3436756.3437015.
- [30] O. R. Ly, K. Gueye, S. Ouya, y G. Mendy, “Collaborative platform for supervision and advice of agricultural engineer to farmers thanks to IoT and WebRTC”, en *The 2021 9th International Conference on Computer and Communications Management (ICCCM '21)*, New York, NY, USA,: Association for Computing Machinery, 2021, pp. 143–148. doi: 10.1145/3479162.3479184.
- [31] S. Pigozzi, F. Faenza, y C. Canali, “Sophon: An Extensible Platform for Collaborative Research”, *Practice and Experience in Advanced Research Computing*, jul. 2022, doi: 10.1145/3491418.
- [32] S. Gebeyehu y H. Twinomurinzi, “A Collaborative Consumption Digital Platform for Government Organizations using Design Science”, *Digital Government: Research and Practice*, vol. 3, núm. 1, ene. 2022, doi: 10.1145/3510615.

- [33] R. Yusri, A. Abusitta, y E. Aïmeur, “Teens-Online: A Game Theory-Based Collaborative Platform for Privacy Education”, *Int J Artif Intell Educ*, vol. 31, núm. 4, pp. 726–768, dic. 2021, doi: 10.1007/s40593-020-00224-0.
- [34] G. Caniglia, “Cast: A context-aware collaborative storytelling platform”, en *Conference on Human Factors in Computing Systems - Proceedings*, Honolulu, HI, USA: Association for Computing Machinery, abr. 2020. doi: 10.1145/3334480.3382966.
- [35] R. Dooley, S. R. Brandt, y J. Fonner, “The agave platform: An open, science-as-a-service platform for digital science”, en *PEARC '18: Practice and Experience in Advanced Research Computing*, Pittsburgh, PA, USA: Association for Computing Machinery, jul. 2018. doi: 10.1145/3219104.3219129.
- [36] R. Ge y H. Sun, “Crowd E-Home—A Crowd-Based Collaborative Decoration Platform”, *International Journal of Crowd Science*, vol. 6, núm. 3, pp. 135–141, sep. 2022, doi: 10.26599/IJCS.2022.9100017.
- [37] N. Zabidi, W. Wang, y D.-L. Xu, “Collaborative supporting tool: Integrating social media and mobile groupware into an integrated learning environment”, en *Proceedings of the 24th International Conference on Automation & Computing*, Newcastle Upon Tyne, UK: Institute of Electrical and Electronics Engineers, sep. 2018, pp. 1–6. doi: 10.23919/IConAC.2018.8749076.
- [38] L. Negrini, S. R. R. Mury, M. Skweres, y N. Kollegger, “Demonstration of Roteco - The robotic teacher community: An informative, collaborative platform with a unique community ecosystem”, en *FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education*, St. Gallen, Switzerland: Association for Computing Machinery, jun. 2021. doi: 10.1145/3466725.3466767.
- [39] T. C. Pargman, S. Joshi, y U. Wehn, “Experimenting with novel forms of computing: The case of the Swedish Citizen Observatory for Water Quality Conservation”, en *Proceedings of ACM LIMITS conference*, Lappeenranta, Finland: Association for Computing Machinery, jun. 2019. doi: 10.1145/3338103.3338111.
- [40] A. Joly, P. Bonnet, A. Affouard, J. C. Lombardo, y H. Goeäu, “Pl@ntNet - My business”, en *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, Mountain View, CA, USA551: Association for Computing Machinery, Inc, oct. 2017, pp. 551–555. doi: 10.1145/3123266.3129312.
- [41] A. Marzouki, S. Mellouli, y S. Daniel, “Spatial, temporal and semantic contextualization of citizen participation”, en *Proceedingsof the 19th Annual International Conference on Digital Government Research*, Delft, Netherlands: Association for Computing Machinery, may 2018. doi: 10.1145/3209281.3209385.
- [42] H. ACM Special Interest Group on Hypertext, Web., y ACM Digital Library., “Citizen Participation: Case Study on Participatory Apps in Germany”, en *WWW*

'18Companion: The 2018 Web Conference Companion, Lyon, France: Association for Computing Machinery, abr. 2018, pp. 915–918. doi: 10.1145/3184558.3191518.

- [43] C. López, R. Farzan, y Y. R. Lin, “Behind the myths of citizen participation: Identifying sustainability factors of hyper-local information systems”, *ACM Trans Internet Technol*, vol. 18, núm. 1, pp. 1–28, nov. 2017, doi: 10.1145/3093892.
- [44] P. Žišt, E. Boaventura, y J. Ramos, “Deployment of a community information platform in Santo Antão island: A case study from Cabo Verde”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, jun. 2021, pp. 160–168. doi: 10.1145/3461564.3461594.
- [45] J. Davies y R. Procter, “Online platforms of public participation: A deliberative democracy or a delusion?”, en *ACM International Conference Proceeding Series*, Association for Computing Machinery, sep. 2020, pp. 746–753. doi: 10.1145/3428502.3428614.
- [46] “Introducción a los conceptos de Angular”, Google LLC. Consultado: el 17 de septiembre de 2022. [En línea]. Disponible en: <https://docs.angular.lat/guide/architecture>
- [47] “FEATURES & BENEFITS”, Google, LLC. Consultado: el 19 de septiembre de 2022. [En línea]. Disponible en: <https://angular.io/features>
- [48] Django Software Foundation, “Why Django?”, django The web framework for perfectionists with deadlines. Consultado: el 24 de septiembre de 2022. [En línea]. Disponible en: <https://www.djangoproject.com/start/overview/>
- [49] “What is PostgreSQL?”, The PostgreSQL Global Development Group. Consultado: el 6 de septiembre de 2022. [En línea]. Disponible en: <https://www.postgresql.org/about/>
- [50] “8.4. Binary Data Types”, The PostgreSQL Global Development Group. Consultado: el 25 de octubre de 2022. [En línea]. Disponible en: <https://www.postgresql.org/docs/current/datatype-binary.html>
- [51] J. Sutherland, *Scrum Handbook*. Scrum Training Institute, 2010. [En línea]. Disponible en: www.scrumtraininginstitute.com
- [52] MkDocs, “REST, Hypermedia & HATEOAS”. Consultado: el 10 de mayo de 2023. [En línea]. Disponible en: <https://www.django-rest-framework.org/topics/rest-hypermedia-hateoas/>
- [53] G. van Rossum, B. Warsaw, y A. Coghlan, “PEP 8 – Style Guide for Python Code”. Consultado: el 24 de noviembre de 2023. [En línea]. Disponible en: <https://peps.python.org/pep-0008/>

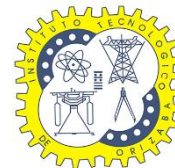
- [54] “Cómo usar OAuth 2.0 para acceder a las API de Google”, Google, LLC. Consultado: el 16 de noviembre de 2023. [En línea]. Disponible en: <https://developers.google.com/identity/protocols/oauth2?hl=es-419>
- [55] “sakai-ng”. Consultado: el 31 de agosto de 2023. [En línea]. Disponible en: <https://github.com/primefaces/sakai-ng>
- [56] “HttpClient”, Google, LLC. Consultado: el 14 de noviembre de 2023. [En línea]. Disponible en: <https://angular.io/api/common/http/HttpClient>
- [57] “Pipe”, Google, LLC. Consultado: el 15 de noviembre de 2023. [En línea]. Disponible en: <https://angular.io/api/core/Pipe>
- [58] “DomSanitizer”, Google, LLC. Consultado: el 15 de noviembre de 2023. [En línea]. Disponible en: <https://angular.io/api/platform-browser/DomSanitizer>



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Orizaba

División de Estudios de Posgrado e Investigación

Maestría en Sistemas Computacionales

TÍTULO:

Artefactos de Ingeniería de Software.

PRESENTADO POR:

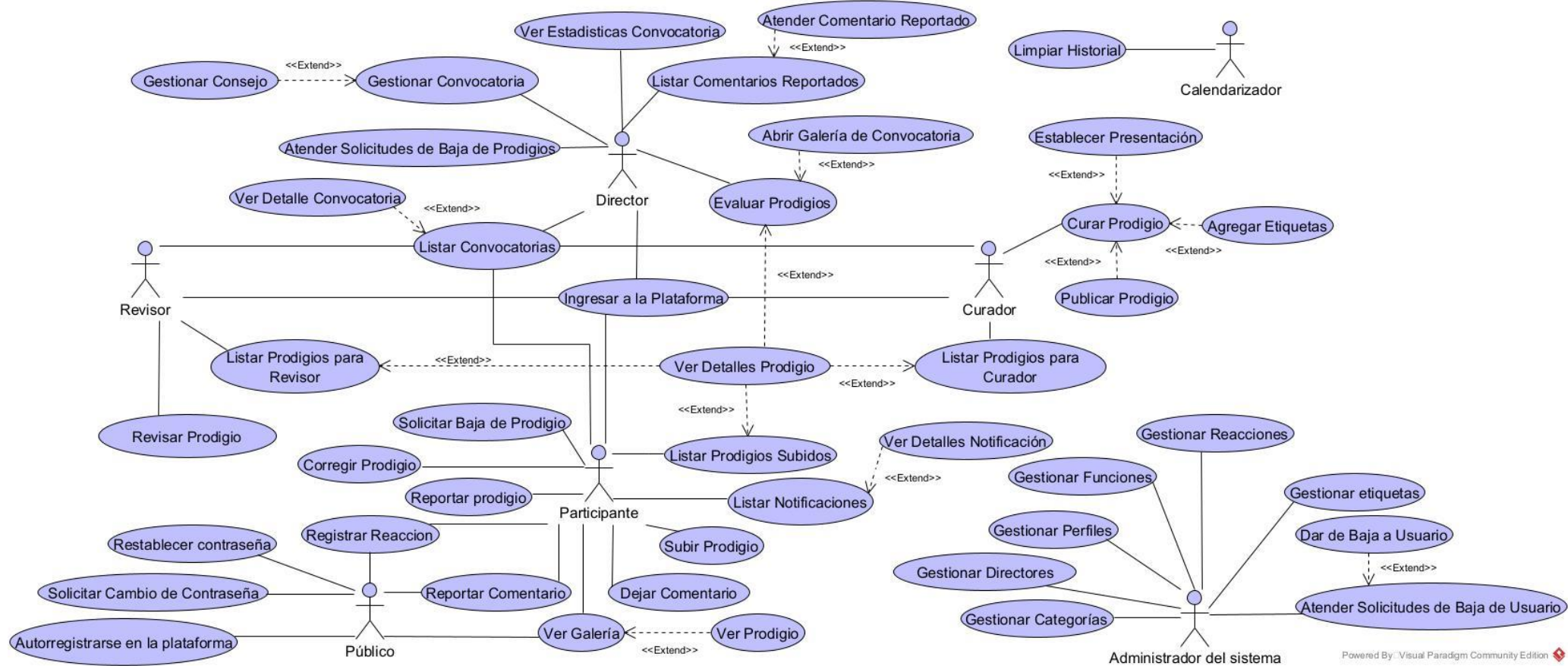
I.S.C. Adolfo Meza Romero M15011191

Índice

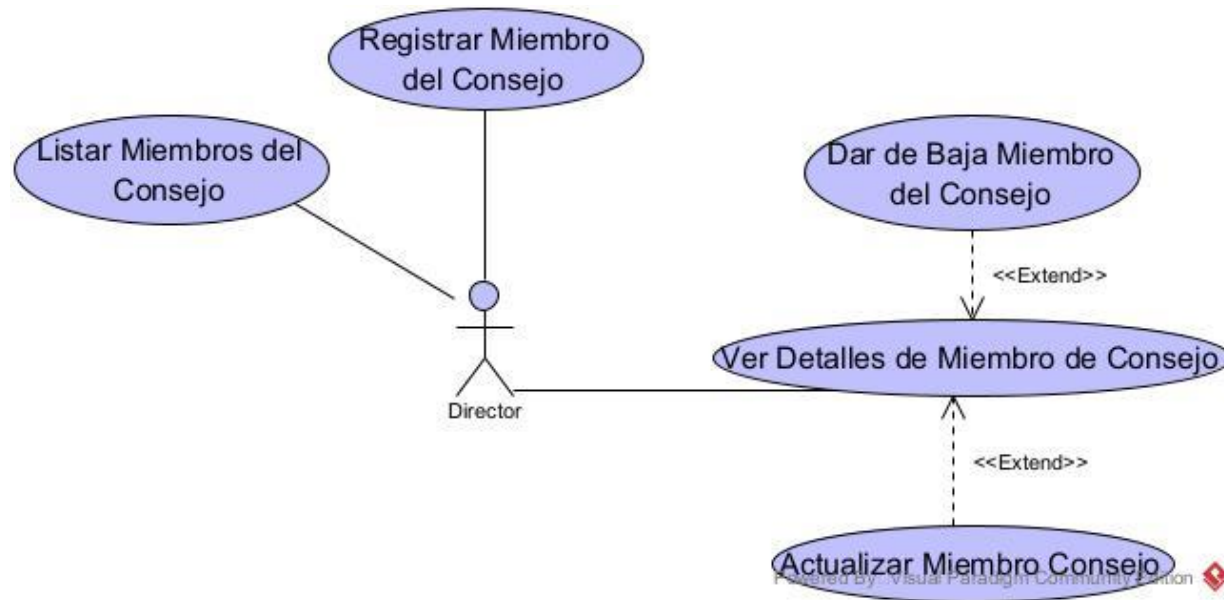
Diagramas de Casos de Uso	1
Diagramas de Casos de uso General	1
Sub diagrama de Casos de uso para director	2
Sub diagrama de Casos de uso para director	2
Diagramas de Clases del Sistema	3
Diagramas Estados	4
Diagrama de Estados de Prodigio	4
Diagrama de Estados de Convocatoria	5
Diagrama de Estados de Comentario.....	5
Módulos de la Plataforma.....	6
Esquema de la Base de Datos.....	7

Diagramas de Casos de Uso

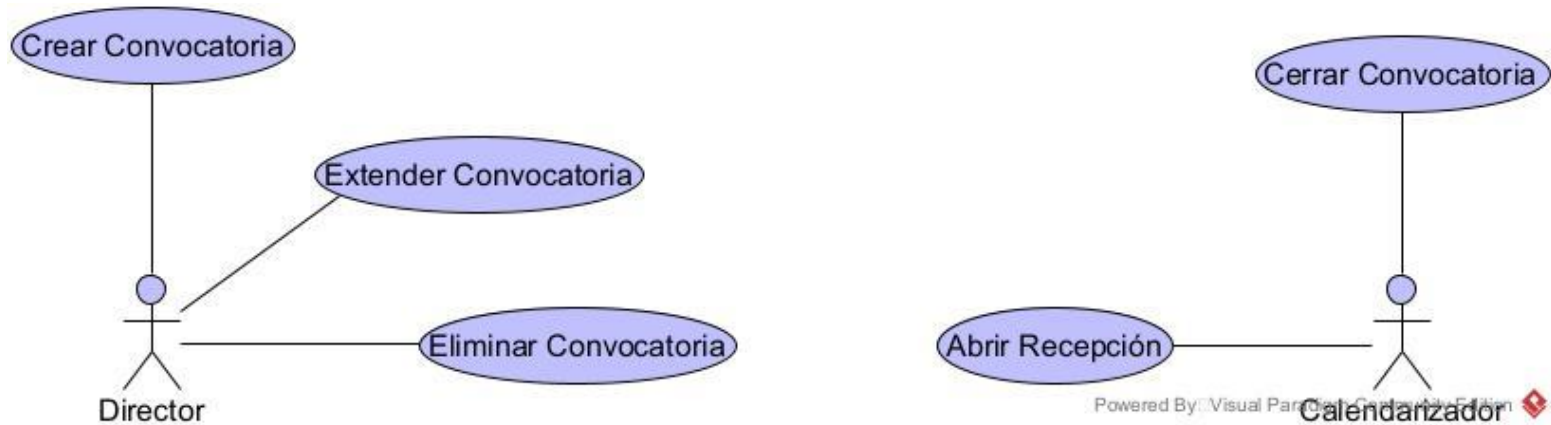
Diagramas de Casos de uso General



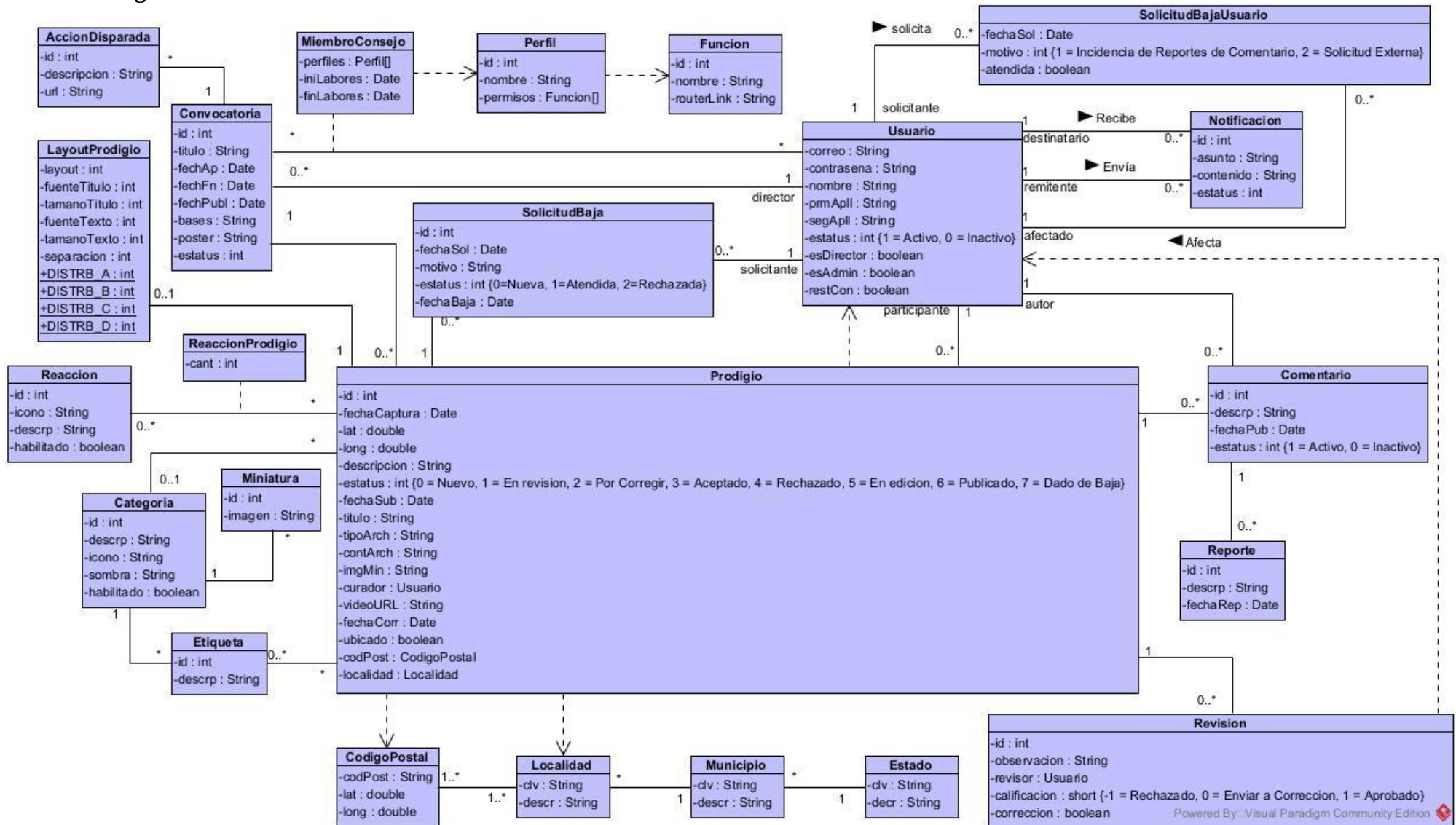
Sub diagrama de Casos de uso para director



Sub diagrama de Casos de uso para director



Diagramas de Clases del Sistema



Diagramas Estados

Diagrama de Estados de Prodigio

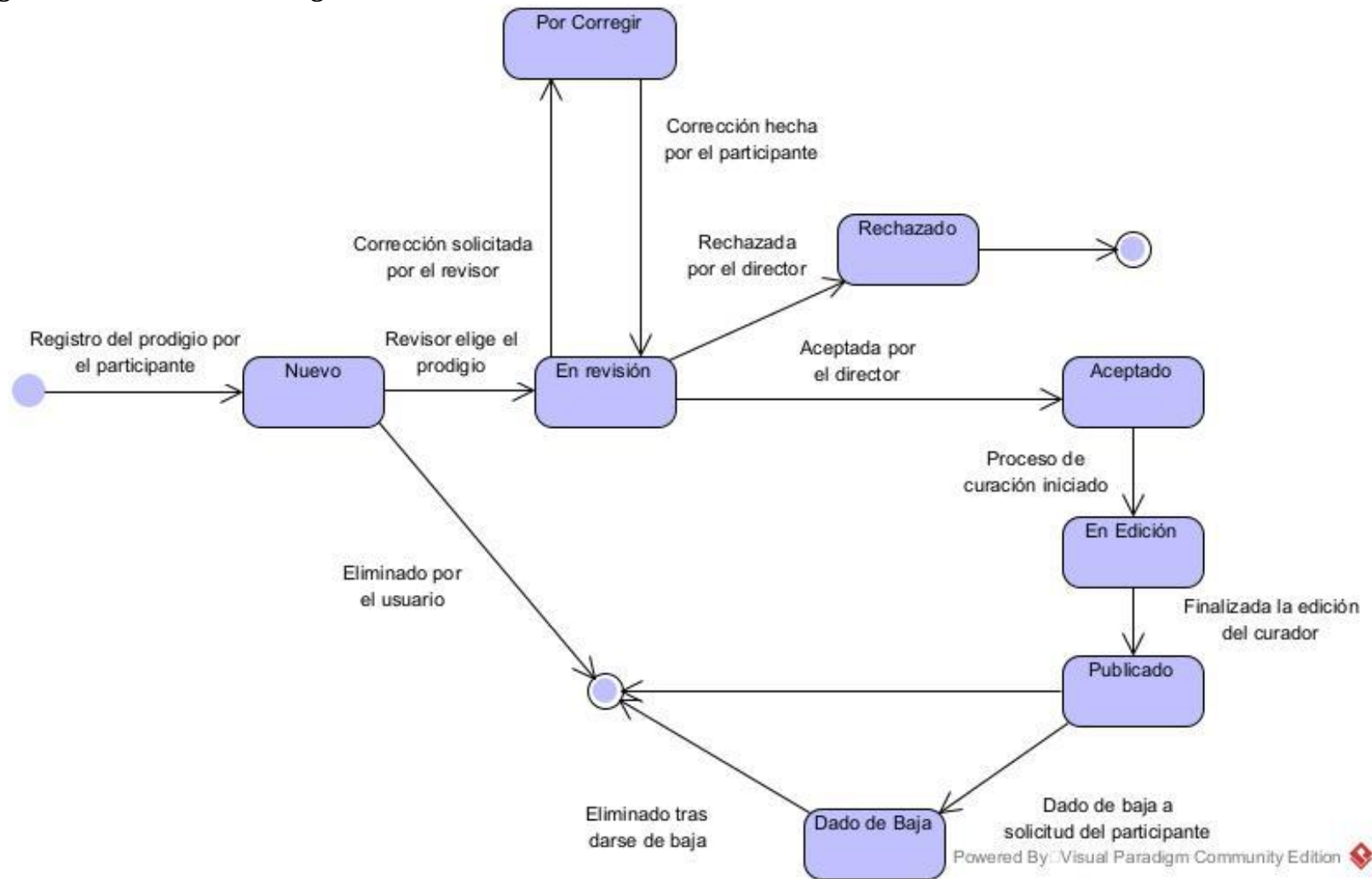


Diagrama de Estados de Convocatoria

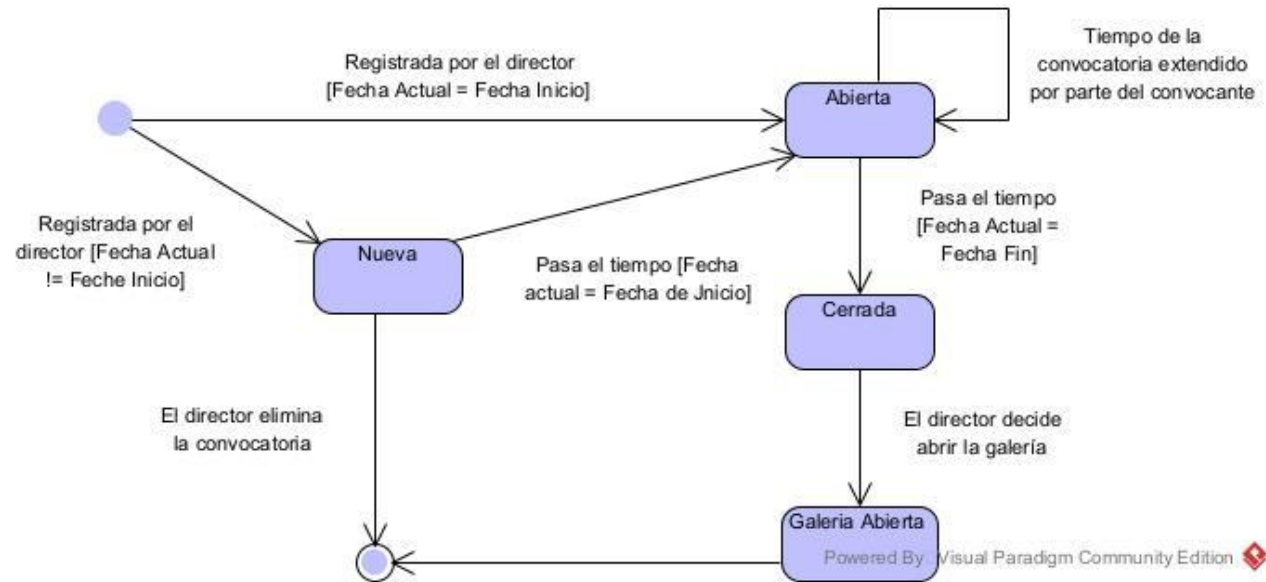
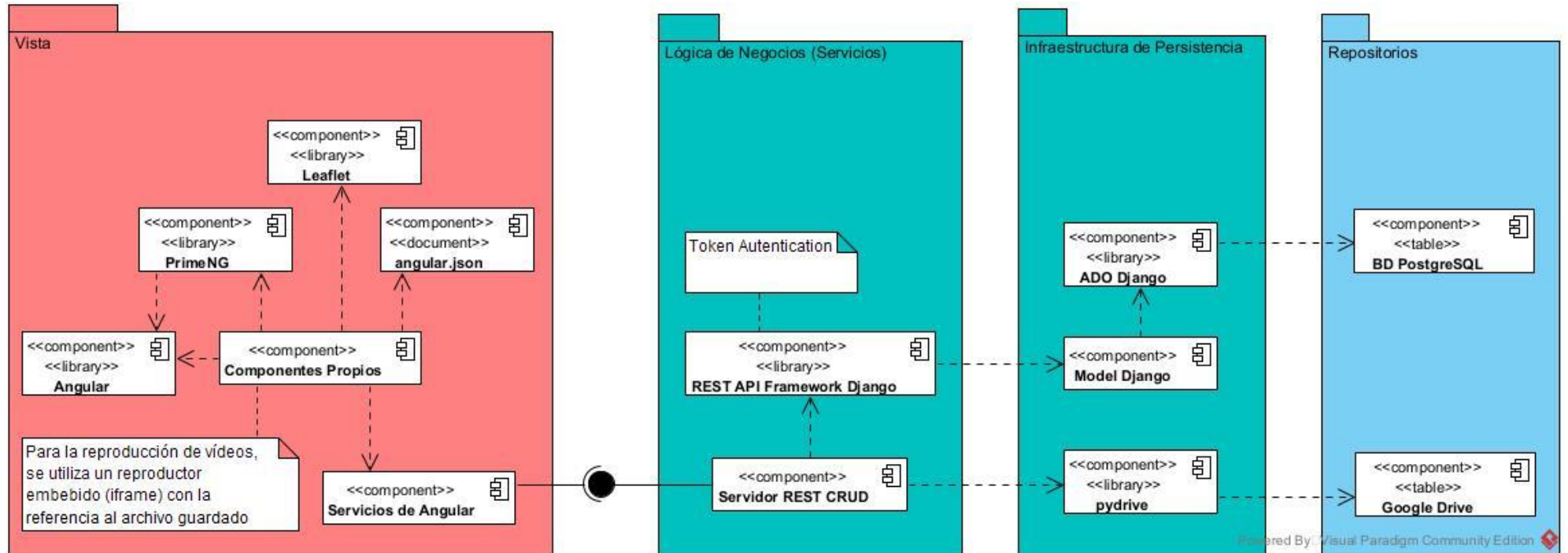


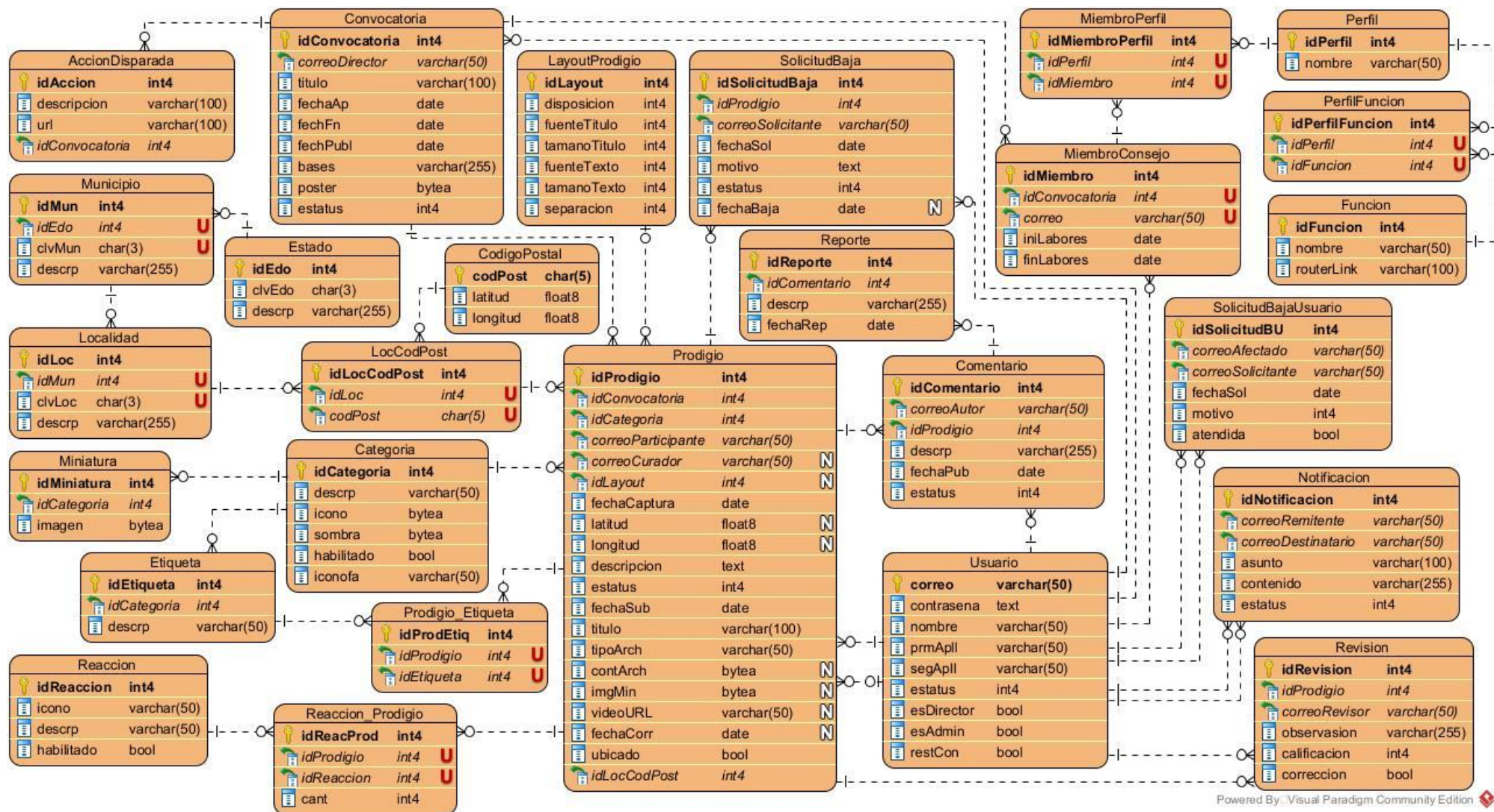
Diagrama de Estados de Comentario



Módulos de la Plataforma



Esquema de la Base de Datos



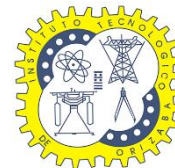
Powered By: Visual Paradigm Community Edition



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Orizaba

División de Estudios de Posgrado e Investigación

Maestría en Sistemas Computacionales

TÍTULO:

Diagrama de Actividades

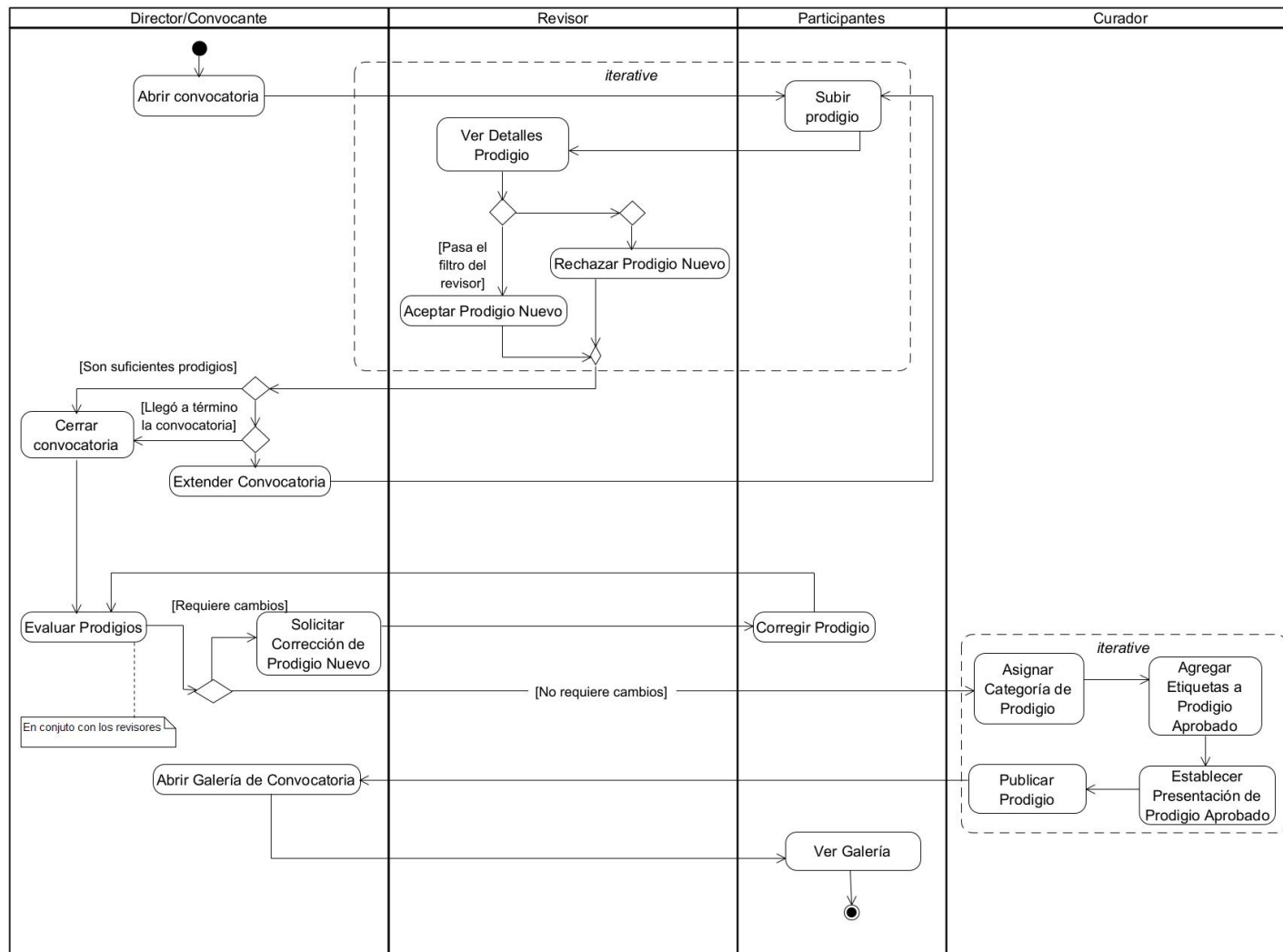
PRESENTADO POR:

I.S.C. Adolfo Meza Romero M15011191

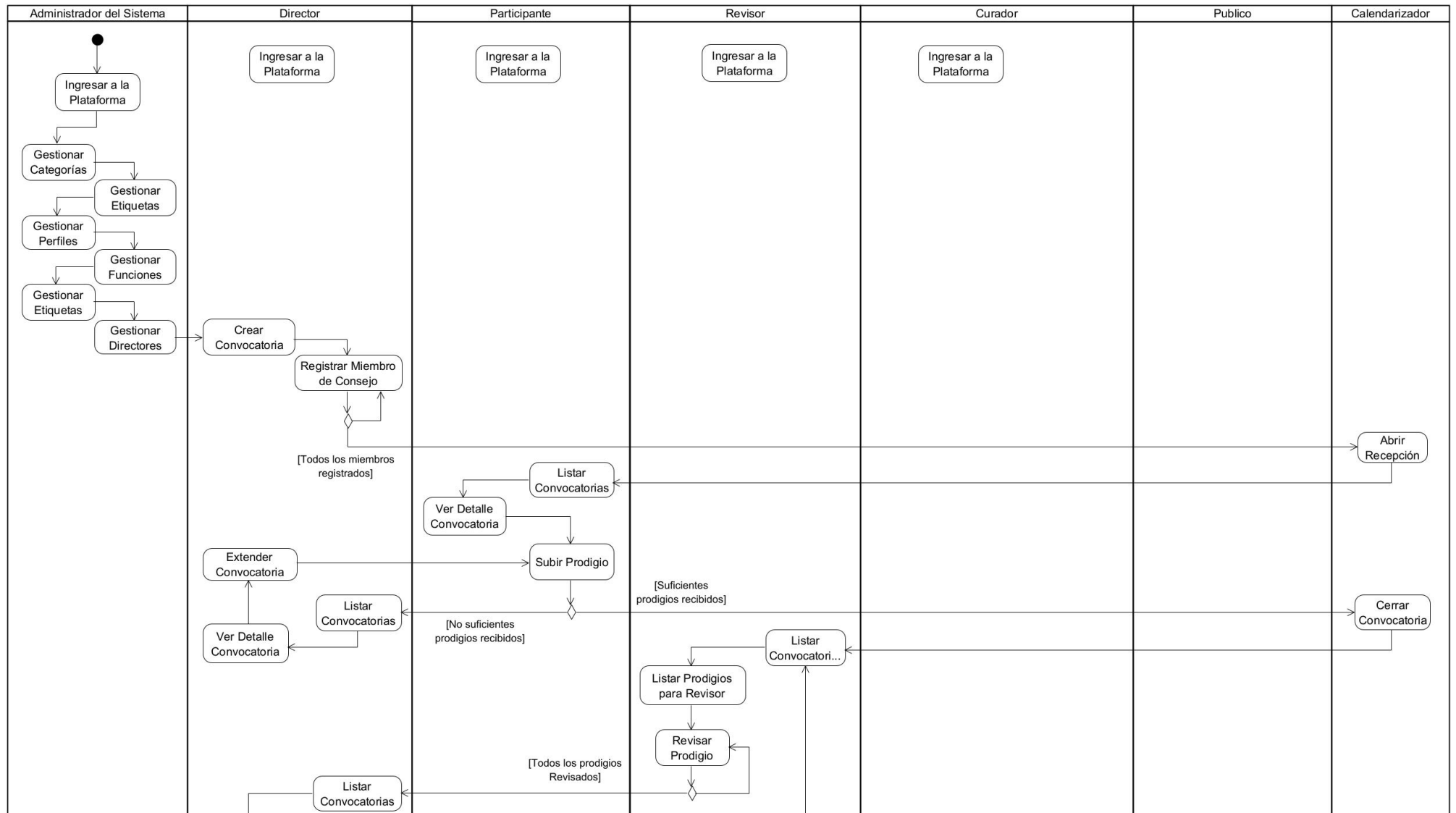
Índice

Diagramas de Actividades.....	1
Proceso del Negocio.....	1
Proceso de la Plataforma.....	2

Proceso del Negocio



Proceso de la Plataforma

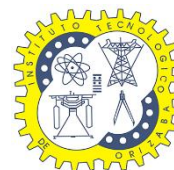




EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Orizaba

División de Estudios de Posgrado e Investigación

Maestría en Sistemas Computacionales

TÍTULO:

Narrativas de Casos de Uso

PRESENTADO POR:

I.S.C. Adolfo Meza Romero M15011191

Índice

Ver Galería.....	1
Ver Prodigio	4
Reportar Comentario	6
Autorregistrarse en la plataforma	7
Ingresar a la plataforma	8
Reportar Prodigio.....	9
Dejar Comentario	10
Listar Prodigios Subidos	11
Ver Detalles Prodigio.....	12
Solicitar Baja de Prodigio	13
Corregir Prodigio	14
Listar Convocatorias	16
Ver Detalle Convocatoria.....	17
Subir Prodigio	18
Listar Notificaciones.....	21
Ver Detalles Notificación.....	22
Listar Prodigios para Revisor	23
Revisar Prodigio	24
Listar Prodigios para Curador.....	26
Curar Prodigio	27
Agregar Etiquetas.....	29
Establecer Presentación	31
Publicar Prodigio	33
Crear Convocatoria.....	34
Listar Miembros del Consejo	35
Registrar Miembro del Consejo	36
Ver Detalle Miembro del Consejo	39
Eliminar Miembro del Consejo	40
Actualizar Miembro del Consejo	41
Evaluar Prodigios	42
Abrir Galería.....	45

Listar Comentarios Reportados	45
Atender Comentario Reportado.....	47
Atender Solicitudes de Baja de Prodigios.....	50

Caso de Uso	Ver Galería	
Objetivo	Mostrar los prodigios subidos en la plataforma, organizándolos por distintas categorías y etiquetas para su fácil ubicación.	
Actor Principal	Público, Participante, Director, Revisor, Curador, Administrador	
Precondiciones	Ninguna	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Se dirige a la pantalla inicial de la plataforma (Mapa de Prodigios). En caso de dirigirse a una de las categorías del menú, ver Curso Alterno 1. En caso de dirigirse a la opción de Convocatorias en el menú, ver Curso alternativo 2.		2. Carga el formulario de filtrado de prodigios (nombre, municipio, localidad, categorías y/o tipos) y el mapa de prodigios (Con la búsqueda por defecto).
3. Modifica los filtros del formulario en función de lo que le interese buscar. 4. Da clic en el botón Filtrar.		5. Busca en la base de datos los prodigios que coincidan con los filtros indicados. En caso de presentar un error, ver Curso de Excepción 1. 6. Pinta en el mapa las ubicaciones de los prodigios obtenidos con un pin. Al pasar el ratón sobre cada pin, este mostrará la información del prodigio que representa. Al dar clic en el pin, se redirigirá a la página del prodigio que representa.
Curso Alterno 1		
Actor		Sistema
1. Selecciona una de las categorías mostradas en el menú.		2. Redirige el visualizador web a la pantalla de la Categoría Seleccionada. 3. Busca en la base de datos las etiquetas asociadas a la categoría seleccionada. En caso de presentar un error, ver Curso de Excepción 2. 4. Busca en la base de datos los prodigios que correspondan a dicha categoría. En caso de presentar un error, ver Curso de Excepción 3. 5. Muestra en pantalla un checkbox por cada etiqueta listada. 6. Muestra en pantalla una cuadrícula de 4x4 con los primeros 16 prodigios listados. Cada uno aparecerá con su miniatura y la información de este, junto a su enlace dentro de la plataforma. En la parte inferior.

	muestra la paginación para acceder a los demás elementos listados.
Curso Alterno 2	
Actor	Sistema
1. Selecciona la opción Convocatorias, en el menú.	2. Redirige el visualizador web a la pantalla de Convocatorias Recientes. 3. Busca en la base de datos las convocatorias más recientes. En caso de presentar un error, ver Curso de Excepción 4. 4. Muestra en pantalla una cuadrícula de 4 x2 con los elementos encontrados. Cada uno aparecerá con su miniatura y la información de este, junto a su enlace dentro de la plataforma.
5. Se dirige a uno de los elementos y da clic en su enlace.	6. Redirige el visualizador web hacia la pantalla de Galería de la Convocatoria. 7. Busca en la base de datos los Prodigios asociados a la Convocatoria seleccionada. En caso de presentar un error, ver Curso de Excepción 5. 8. Muestra en pantalla una cuadrícula de 4x4 con los primeros 16 prodigios listados. Cada uno aparecerá con su miniatura y la información de este, junto a su enlace dentro de la plataforma. En la parte inferior, muestra la paginación para acceder a los demás elementos listados.
Curso de Excepción 1	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudieron recuperar la lista de prodigios indicada”.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de etiquetas de la categoría”. 2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.
Curso de Excepción 3	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de prodigios de la categoría”.

	2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.
Curso de Excepción 4	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de convocatorias recientes”. 2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.
Curso de Excepción 4	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de prodigios de esta convocatoria”. 2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Ver Prodigio	
Objetivo	Mostrar el contenido de un prodigio seleccionado por el usuario en una presentación adecuada para la galería.	
Actor Principal	Público, Participante, Director, Revisor, Curador, Administrador	
Precondiciones	Haber pasado por el caso de uso Ver Galería.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor	Sistema	
1. Da clic en uno de los prodigios listados en pantalla, o bien, en uno de los prodigios representados en el mapa.	2. Redirige el visualizador web hacia la pantalla de Ver Prodigio. 3. Busca en la base de datos el prodigio seleccionado. En caso de presentar un error, ver Curso de Excepción 1. 4. Busca en la base de datos los comentarios asociados al prodigio seleccionado. En caso de presentar un error, ver Curso de Excepción 2. 5. Muestra en la pantalla el contenido del prodigio. Debajo del prodigio, anexa un botón para compartir el prodigio y, si hay una sesión iniciada, un botón para Reportar el Prodigio. Por debajo, si hay una sesión iniciada y está accediendo como participante, muestra un área de texto para capturar y publicar un comentario. En la parte inferior, enlista los comentarios asociados al prodigio. Del lado izquierdo de cada comentario muestra un botón para reportar comentario.	
Curso Alterno 1		
Actor	Sistema	
Curso de Excepción 1		
Actor	Sistema	
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la información del prodigio seleccionado”. 2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.	
Curso de Excepción 2		
Actor	Sistema	
	1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de comentarios del prodigio”.	

		2. Redirige el visualizador web a la pantalla de Mapa de Prodigios.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Reportar Comentario	
Objetivo	Permitir levantar un reporte a un comentario que infrinja las normativas de comportamiento de la plataforma.	
Actor Principal	Público, Participante.	
Precondiciones	Haber pasado por el caso de uso Ver Prodigio.	
Postcondiciones	Un nuevo registro de Reporte en la base de datos.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla del prodigio, en la sección de comentarios, da clic en el botón de Reportar (reconocible por el ícono de una banderita) de alguno de los comentarios.		2. Redirige el visualizador web a la pantalla de Reportar Comentario. 3. Carga el formulario de reporte de comentario.
4. Captura en el formulario el motivo por el cual se reporta el comentario. 5. Da clic en el botón “Reportar Comentario”.		6. Guarda en la base de datos un nuevo registro Reporte de comentario. Si ocurre algún error, ir al Curso de Excepción 1. 7. Muestra una alerta con el mensaje “Reporte realizado exitosamente”. 8. Redirige el visualizador web a la pantalla de Ver Prodigio.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo realizar el reporte del comentario, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Autorregistrarse en la plataforma	
Objetivo	Permitir a cualquier visitante crear una cuenta de usuario para acceder a la plataforma.	
Actor Principal	Público	
Precondiciones	Ninguna	
Postcondiciones	Un nuevo registro de Usuario en la Base de Datos	
Flujo Básico		
Actor		Sistema
1. En el menú superior de la pantalla, sin haber iniciado sesión, selecciona la opción de Iniciar Sesión.		2. Redirige el visualizador a la pantalla Iniciar Sesión.
4. Da clic en el enlace indicado para Registrarse.		3. Carga el formulario de inicio de sesión.
7. Captura la información solicitada en el Formulario.		5. Redirige el visualizador a la pantalla de Registro de usuario.
8. Da clic en el botón Crear Cuenta.		6. Carga el formulario de Registro de Usuario (Nombre, Primer Apellido, Segundo Apellido, Correo, Contraseña).
		9. Guarda en la base de datos un nuevo registro de Usuario con los datos capturados en el Formulario. Si ocurre un error, ir al Curso de Excepción 1.
		10. Muestra una alerta con el mensaje “Cuenta registrada”.
		11. Redirige la pantalla de inicio de Sesión.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo dar de alta al usuario, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Ingresar a la plataforma	
Objetivo	Permitir a los usuarios iniciar una sesión en la plataforma.	
Actor Principal	Participante, Director, Revisor, Curador, Administrador	
Precondiciones	El usuario debe poseer una cuenta en la plataforma.	
Postcondiciones	El usuario habrá iniciado sesión en la plataforma.	
Flujo Básico		
Actor		Sistema
1. Sin haber iniciado sesión, dirigirse a la opción de inicio de sesión del menú superior.		2. Redirige el visualizador a la pantalla de inicio de sesión. 3. Carga el formulario de inicio de sesión (Correo, Contraseña).
4. Captura las credenciales de usuario en el formulario. 5. Da clic en el botón de “Iniciar Sesión”.		6. Busca en la base de datos un Usuario que coincida con las credenciales ingresadas. Si ocurre un error, ver Curso de Excepción 1. 7. Si no encuentra ninguna coincidencia, ver Curso Alterno 1. 8. Tras encontrar al usuario, redirige la pantalla hacia el panel de participante.
Curso Alterno 1		
Actor		Sistema
		1. Lanza una alerta con el mensaje “No se reconocen las credenciales del usuario”.
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo iniciar sesión, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Reportar Prodigio	
Objetivo	Permitir a los usuarios reportar un prodigio que infrinja alguna de las normas de la plataforma	
Actor Principal	Participante	
Precondiciones	Haber pasado por el caso de uso Ver Prodigio.	
Postcondiciones	Se generará un nuevo registro de SolicitarBaja.	
Flujo Básico		
Actor		Sistema
1. Da clic en el botón Reportar Prodigio		2. Redirige el visualizador web a la pantalla de Reportar Prodigio. 3. Carga el formulario de Reportar Prodigio.
4. Captura el motivo de Reporte de prodigio, 5. Da clic en el botón Reportar Prodigio.		6. Registra en la base de datos un nuevo objeto SolicitudBaja con los datos proporcionados en el formulario. Si ocurre un error, ir al Curso de Excepción 1. 7. Muestra en pantalla una alerta con el mensaje “El reporte se realizó con éxito”. 8. Redirige el visualizador web a la pantalla de Ver Prodigio.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo realizar el reporte del prodigio, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Dejar Comentario	
Objetivo	Permitir al Participante interactuar con un prodigio de su interés, dejando un comentario.	
Actor Principal	Participante	
Precondiciones	Haber pasado por el caso de uso Ver Prodigio.	
Postcondiciones	Se generará un nuevo registro Comentario en la base de Datos.	
Flujo Básico		
Actor		Sistema
1. Se dirige a la sección del formulario de Comentarios en la pantalla de Ver Prodigio. 2. Captura la descripción de su comentario. 3. Da clic en el botón “Publicar Comentario”		4. Guarda un nuevo objeto Comentario en la base de datos. Si ocurre un error, ir al Curso de Excepción 1. 5. En pantalla, agrega el comentario realizado a la lista de comentarios del prodigio.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo registra el comentario, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Listar Prodigios Subidos	
Objetivo	Enlistar los prodigios que ha subido el participante.	
Actor Principal	Participante	
Precondiciones	Haber iniciado sesión en la plataforma.	
Postcondiciones	Ninguna.	
Flujo Básico		
Actor		Sistema
1. Desde el menú lateral del panel de Participante, selecciona la opción “Mis prodigios”.		2. Redirige el visualizador web hacia la pantalla de Listar Prodigios Subidos. 3. Busca en la base de datos todos los prodigios que hayan sido creados por el usuario en sesión. Si ocurre un error, ver Curso de Excepción 1. 4. Pinta una tabla con los prodigios encontrados, cada uno con las opciones de acción “Ver Prodigio” y “Corregir” (Esta última en caso de que el estatus del prodigio sea “Por Corregir”).
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de prodigios subidos”. 2. Redirige el visualizador web a la pantalla de inicio del panel del Participante.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Ver Detalles Prodigio	
Objetivo	Permitir al Participante ver los detalles del prodigio seleccionado.	
Actor Principal	Participante, Revisor, Curador.	
Precondiciones	Haber iniciado sesión en la plataforma, haber pasado por cualquiera de los casos de uso Listar Prodigios Subidos, Listar Prodigios para Revisor, Listar Prodigios para Curador.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Desde la lista de prodigios (Independientemente del caso de uso en el que se encuentre), da clic en la opción “Ver Prodigio”.		2. Redirige el visualizador web a la pantalla Ver Detalles Prodigio. 3. Busca en la base de datos el prodigio seleccionado. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra toda la información que incluye el prodigio en un formulario (Titulo, Categoría, Fecha de Captura, Tipo de Archivo, Municipio, Localidad, Ubicación, Archivo, Descripción). Por defecto, esto campos deben encontrarse deshabilitados. Para el participante, en la parte inferior del formulario, aparecerá un botón que permitirá solicitar la eliminación del prodigio.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la información del prodigio seleccionado, inténtelo más tarde”. 2. Redirige el visualizador web a la pantalla de Listar Prodigios Subidos.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Solicitar Baja de Prodigio	
Objetivo	Permitir al Participante solicitar la baja de un prodigio que haya subido.	
Actor Principal	Participante.	
Precondiciones	Haber iniciado sesión en la plataforma, haber pasado por el caso de uso Ver Detalles Prodigio.	
Postcondiciones	Se generará un nuevo registro de SolicitudBaja en la base de Datos.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Ver Detalle Prodigio (Participante), da clic en el botón “Solicitar Eliminación”.		2. Registra en la base de datos un nuevo registro SolicitudBaja, con la fecha actual como fecca de solicitud, y como motivo “Baja a petición del Autor”. Si ocurre un error, ver Curso de Excepción 1. 3. Muestra en pantalla una alerta con el mensaje “Solicitud de Baja realizada exitosamente”. 4. Redirige el visualizador web a la pantalla Ver Prodigios Subidos.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo realizar la Solicitud de Baja, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Corregir Prodigio	
Objetivo	Permitir al Participante corregir las observaciones realizadas sobre un prodigio que haya subido.	
Actor Principal	Participante	
Precondiciones	Haber pasado por el caso de uso Listar Prodigios Subidos.	
Postcondiciones	El registro de Prodigio actualizado y con estatus “En revisión”.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Prodigios Subidos, seleccionar la opción “Corregir”.		2. Redirige el visualizador web a la pantalla de Corregir Prodigio. 3. Busca en la base de datos el prodigio seleccionado. Si ocurre un error, ver Curso de Excepción 1. 4. Busca en la base de datos las observaciones asociadas al prodigio encontrado. Si ocurre un error, ver Curso de Excepción 2. 5. Carga el formulario de Prodigio, lleno con los atributos del prodigio encontrado (Algunos campos se encontrarán deshabilitados a excepción de Título, Categoría, y Descripción). 6. Carga las observaciones de los revisores en una tabla.
7. Modifica los campos habilitados en función de las observaciones realizadas por los revisores. 8. Da clic en el botón “Re Enviar Prodigios”.		9. Guarda los cambios realizados en la base de datos. Si ocurre un error, ver Curso de Excepción 3. 10. Muestra una alerta con el mensaje “Cambios realizados exitosamente”. 11. Redirige el navegador hacia la pantalla de Listar Prodigios Subidos.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la información del prodigio seleccionado, inténtelo más tarde”. 2. Redirige el visualizador web a la pantalla de Listar Prodigios Subidos.
Curso de Excepción 2		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudieron recuperar las

		observaciones del prodigio seleccionado, inténtelo más tarde”. 2. Redirige el visualizador web a la pantalla de Listar Prodigios Subidos.
Curso de Excepción 3		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudieron realizar los cambios en el prodigio seleccionado, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Listar Convocatorias	
Objetivo	Permitir al Participante visualizar la lista de convocatorias disponibles en las que puede participar.	
Actor Principal	Público, Participante, Director, Revisor, Curador.	
Precondiciones	Haber iniciado sesión en la plataforma.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Desde el panel de Participante, dar clic en la opción Convocatorias.		2. Redirige el visualizador web a la pantalla de Listar Convocatorias. 3. Si el usuario que accede es Participante o se accede desde la galería, busca en la base de datos las convocatorias que se encuentran abiertas. Si el usuario es director, busca las convocatorias registradas por el Director. Si el usuario es Revisor, busca todas las convocatorias en las que participa como revisor. Si el usuario es Curador, busca todas las convocatorias en las que participa como curador. Si ocurre un error, ver Flujo de Excepción 1. 4. Muestra en pantalla una tabla con las convocatorias encontradas. Se es en el panel del participante cada una con la opción de acción “Ver Detalles”. Si es en el caso del Revisor, Curador o Director, agrega la opción “Ver Prodigios”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la lista de convocatorias, inténtelo más tarde”. 2. Redirige el visualizador web a la pantalla de inicio del panel que se encuentre activo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Ver Detalle Convocatoria	
Objetivo	Permitir al usuario visualizar los detalles de una convocatoria seleccionada.	
Actor Principal	Participante, Director, Revisor, Curador.	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Convocatorias.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Convocatorias, selecciona la opción Ver Detalles.		2. Redirige el visualizador web a la pantalla Detalles de Convocatoria. 3. Busca en la base de datos el registro de la convocatoria seleccionada. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla la información de la convocatoria seleccionada. 5. Muestra en la parte inferior los botones de acción de la convocatoria. En el caso de Participante, la única acción es “Participar”. En el caso de Revisor y Curador, la acción es “Ver Prodigios”. En el caso del director, se presentarán las acciones de “Extender Fecha de Terminación”, “Eliminar Convocatoria” y “Gestionar Consejo”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la información de la convocatoria seleccionada”. 2. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Subir Prodigio	
Objetivo	Permitir al participante compartir los prodigios ambientales que identifique a través de una de las convocatorias registradas en la plataforma.	
Actor Principal	Participante	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Ver Detalle Convocatoria.	
Postcondiciones	Un nuevo registro de prodigio, con estatus Nuevo, guardado en la base de datos.	
Flujo Básico		
Actor	Sistema	
1. Desde la pantalla Ver Detalle Prodigio, dar clic en el botón “Participar”.	2. Redirige el visualizador web a la pantalla de Subir Prodigio. 3. Busca en la base de datos las categorías disponibles. Si ocurre un error, ver curso de Excepción 1. 4. Busca en la base de datos todos los municipios del estado de Veracruz. Si ocurre un error, ver curso de Excepción 2. 5. Carga el formulario de captura de prodigio (Título, Fecha de Captura, Tipo Archivo, Archivo, Categoría, Municipio, Localidad, Código Postal [Deshabilitado], Omitir Ubicación en el Mapa, Descripción, Latitud [Oculta], Longitud [Oculta]), colocando las categorías y los municipios listados en sus respectivas listas desplegables. 6. Carga un mapa del estado de Veracruz, haciendo uso de leaflet, con un punto desplazable para la ubicación del prodigio.	
7. Captura la información del prodigio. 8. Selecciona un Municipio	9. Busca en la base de datos las localidades que se encuentran dentro del municipio seleccionado. Si ocurre un error, ver curso de Excepción 3. 10. Coloca las localidades encontradas en su respectiva lista desplegable.	
11. Selecciona una localidad	12. Busca en la base de datos los códigos postales que se encuentran dentro de la localidad seleccionada. Si ocurre un error, ver curso de Excepción 4. 13. Coloca los códigos postales encontrados en su respectiva lista desplegable.	

14. Marca la opción Omitir ubicación en el mapa. En el caso contrario, ver Curso Alternativo 1.	15. Habilita el campo Código Postal. 16. Oculta el mapa de Veracruz.
17. Selecciona un código Postal 18. Da clic en el botón Subir Prodigio	19. Guarda en la base de datos un nuevo objeto Prodigio con la información capturada en el formulario. Si ocurre un error, ver curso de excepción 5. 20. Muestra una alerta indicando que el prodigio fue subido con éxito. 21. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso Alternativo 1	
Actor	Sistema
1. Arrastrando el pin en el mapa, ubica el prodigio.	2. Almacena las coordenadas de latitud y longitud identificadas por el mapa.
3. Si requiere reubicar el prodigio, volver al punto 1 del Curso Alternativo 1. 4. Vuelve al punto 18 del Flujo Básico.	
Curso de Excepción 1	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de categorías”. 2. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de Municipios”. 2. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso de Excepción 3	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de localidades”. 2. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso de Excepción 4	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de Códigos Postales”. 2. Redirige el visualizador web a la pantalla de Listar Convocatorias.

Curso de Excepción 5	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo subir el prodigio debido a un error en la base de datos, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Listar Notificaciones	
Objetivo	Permitir al usuario ver las notificaciones recibidas.	
Actor Principal	Participante, Director, Revisor, Curador.	
Precondiciones	Haber iniciado sesión en la plataforma.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Desde cualquier panel, en el menú superior, dar clic en la opción Notificaciones.		2. Redirige el visualizador web a la pantalla de Listar Notificaciones. 3. Busca en la base de datos todas las Notificaciones que tienen al usuario en sesión como destinatario, ordenadas de la más reciente a la más antigua. Si ocurre un error, ir al Curso de Excepción 1. 4. En pantalla, muestra una tabla con todas las notificaciones encontradas, cada una con la opción de acción Ver Detalles.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de notificaciones”. 2. Redirige el visualizador web a la pantalla de inicio del panel activo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Ver Detalles Notificación	
Objetivo	Mostrar al usuario el contenido de una notificación recibida.	
Actor Principal	Participante, Director, Revisor Curador.	
Precondiciones	Haber iniciado sesión en la plataforma, haber ejecutado el caso de uso Listar Notificaciones.	
Postcondiciones	Ninguna.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Notificaciones, dar clic en la acción Ver Detalles.		2. Redirige el visualizador web a la pantalla Ver Detalles Notificación. 3. Busca en la base de datos el registro correspondiente a la Notificación seleccionada. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla los detalles de la notificación (Asunto, Remitente y Contenido).
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo recuperar la información de esta notificación”. 2. Redirige el visualizador a la pantalla de inicio del panel que se encuentre activo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Listar Prodigios para Revisor	
Objetivo	Listar los prodigios que han sido subidos dentro de una convocatoria específica para su posterior evaluación por parte del Revisor.	
Actor Principal	Revisor	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Convocatorias.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Ver Prodigios” de una de las convocatorias de la lista de convocatorias mostrada.		2. Redirige el visualizador web a la pantalla de Listar Prodigios para Revisor. 3. Busca en la base de datos todos los prodigios asociados a la convocatoria seleccionada, que se encuentren con estatus Nuevo o En revisión. Si ocurre un error, ir al Curso de Excepción 1. 4. En pantalla muestra una tabla que enliste los elementos encontrados, cada uno con la opción “Ver Prodigio”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de prodigios”. 2. Redirige el visualizador web a la pantalla de inicio del panel de Revisor.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Revisar Prodigio	
Objetivo	Permitir al revisor evaluar el contenido de un prodigio subido en una convocatoria específica.	
Actor Principal	Revisor	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Prodigios para Revisor.	
Postcondiciones	Un nuevo registro de Revision guardado en la base de datos.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción Ver Prodigio en una de las tuplas de la tabla de Prodigios.		2. Redirige el visualizador web a la pantalla de Revisar Prodigio. 3. Busca en la base de datos la información del prodigio seleccionado. Si ocurre un error, ir al Curso de Excepción 1. 4. Busca en la base de datos las revisiones realizadas sobre el prodigio seleccionado. Si ocurre un error, ir al Curso de Excepción 2. 5. Muestra en pantalla los datos del prodigio obtenido (Titulo, Categoría, Fecha de Captura, Tipo de Archivo, Municipio, Localidad, Ubicación, Archivo, Descripción) en un formulario con los campos deshabilitados. 6. En la parte inferior del formulario del Prodigio, muestra el formulario de Revisión (Aprobado/Rechazado, Requiere Corrección?, Observación). Si el usuario en sesión ya ha registrado una observación, ver Curso Alterno1. 7. Muestra en la parte inferior del formulario de Revisión, una tabla con las revisiones ya realizadas al prodigio seleccionado.
8. Captura la información requerida por el Formulario de Revisión. 9. Da clic en el botón “Guardar Revisión”.		10. Almacena en la base de datos la revisión realizada. Si ocurre un error, ver Curso de Excepción 3. 11. Agrega a la tabla de Revisiones, la revisión realizada. 12. Deshabilita el Formulario de Revisiones.
Curso Alterno 1		
Actor		Sistema
		1. Deshabilita el Formulario de Revisión.

	2. Muestra en la parte inferior del formulario de captura de Revisión, una tabla con las revisiones ya realizadas al prodigio seleccionado.
Curso de Excepción 1	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la información del prodigio seleccionado”. 2. Redirecciona el visualizador a la pantalla de Listar Prodigios para Revisor.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de revisiones”. 2. Redirecciona el visualizador a la pantalla de Listar Prodigios para Revisor.
Curso de Excepción 3	
Actor	Sistema
	1. Muestra en pantalla una alerta con el mensaje “No se pudo guardar el registro en la base de datos, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Listar Prodigios para Curador	
Objetivo	Listar los prodigios que han sido subidos dentro de una convocatoria específica para su posterior curaduría por parte del Curador.	
Actor Principal	Curador	
Precondiciones	Haber iniciado sesión en la plataforma, haber ejecutado el caso de uso Listar Convocatorias.	
Postcondiciones	Ninguna.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Ver Prodigios” en una de las convocatorias de la lista.		2. Redirige el visualizador web a la pantalla de Listar Prodigios para Curador. 3. Busca en la base de datos todos los prodigios con estatus Aprobado, que se encuentren asociados a la convocatoria seleccionada. Si ocurre un error, ver Curso de Excepción 1. 4. Dibuja una tabla llena con los prodigios encontrados. Cada prodigio debe presentar la opción “Editar”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra en pantalla una alerta con el mensaje “No se pudo obtener la lista de prodigios”. 2. Redirige el visualizador web a la pantalla de inicio del panel de Curador.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Curar Prodigio	
Objetivo	Corregir, de un prodigio seleccionado, los detalles correspondientes a la ubicación del prodigio en cuestión, así como brindar el acceso a otras opciones como el agregado de etiquetas o el establecer la presentación del prodigio.	
Actor Principal	Curador.	
Precondiciones	Haber ejecutado el caso de uso Listar Prodigios para Curador.	
Postcondiciones	El objeto Prodigio seleccionado, con las coordenadas de su ubicación Actualizadas.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Editar”.		2. Redirige el visualizador web a la pantalla de Curar Prodigio. 3. Busca en la base de datos la información del prodigio seleccionado. Si ocurre un error, ver Curso de Excepción 1. 4. Carga el formulario de captura de prodigio (Título, Fecha de Captura, Tipo Archivo, Archivo, Categoría, Código Postal, Descripción, Archivo, Latitud [Oculto], Longitud [Oculto]) con los campos deshabilitados y llénalo con la información recuperada del prodigio seleccionado. Si el prodigio ya posee una coordenada de Latitud y Longitud, ver Curso Alterno 1. 5. Carga un mapa del estado de Veracruz, por defecto ubicado en la zona correspondiente al código postal del prodigio, con un pin desplazable para la ubicación del prodigio. 6. En la parte inferior, pinta los botones “Agregar Etiquetas”, “Ajustar Presentación” y “Publicar Prodigio”.
7. Ubica el pin en la posición donde se encontró el prodigio.		8. Actualiza los campos Latitud y Longitud en la base de datos con la nueva ubicación del pin.
9. Si requiere volver a cambiar la ubicación del prodigio, volver al paso 7 del Flujo Básico. 10. Da clic en el botón “Guardar Ubicación Nueva”.		11. Actualiza el prodigio seleccionado con la nueva ubicación indicada. Si ocurre un error, ver Curso de Excepción 2. 12. Muestra una alerta con el Mensaje “Se ha actualizado la ubicación del prodigio”. 13. Deshabilita el botón “Guardar Ubicación Nueva”.

	14. Deshabilita el movimiento del pin en el mapa.
Curso Alterno 1	
Actor	Sistema
	1. Carga un mapa del estado de Veracruz, por defecto ubicado en la zona correspondiente al código postal del prodigio, con un pin ubicado en las coordenadas en donde se ubicó el prodigio. 2. Deshabilita el botón “Guardar Ubicación Nueva”. 3. En la parte inferior, pinta los botones “Agregar Etiquetas”, “Ajustar Presentación” y “Publicar Prodigio”.
Curso de Excepción 1	
Actor	Sistema
	1. Muestra una alerta con el mensaje “No se pudo obtener la información del prodigio”. 2. Redirige el visualizador web a la pantalla de Listar Prodigios para Curador.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra una alerta con el mensaje “No se pudo actualizar la ubicación del prodigio, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Agregar Etiquetas	
Objetivo	Permitir al Curador agregar las etiquetas que permitan ubicar al prodigio dentro de la categoría que lo incluye.	
Actor Principal	Curador	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Curar Prodigio.	
Postcondiciones	Se almacenarán nuevos registros que asociarán al prodigio seleccionado con una serie de etiquetas indicadas.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Curar Prodigio, da clic en el botón “Agregar Etiquetas”.		2. Redirige el visualizador web a la pantalla de Agregar Etiquetas. 3. Busca en la base de datos todas las etiquetas asociadas a la categoría a la que pertenece el prodigio. Si ocurre un error, ver Curso de Excepción 1. 4. Busca en la base de datos todas las etiquetas a las que se encuentra asociado el prodigio. Si ocurre un error, ver Curso de Excepción 2. 5. Muestra en pantalla, del lado derecho, una tabla de elementos <i>Drag & Drop</i> que enliste las etiquetas disponibles de la categoría. 6. Muestra en pantalla, del lado izquierdo, una tabla de elementos <i>Drag & Drop</i> que enliste las etiquetas a las que se encuentra asociado el prodigio. 7. Muestra en la parte inferior los botones “Volver a Detalles del prodigio” y “Guardar Cambios”.
8. Si requiere agregar etiquetas al prodigio, arrastra elementos de la tabla de etiquetas disponibles a la tabla de etiquetas asociadas. 9. Si requiere remover etiquetas del prodigio, arrastra elementos de la tabla de etiquetas asociadas a la tabla de etiquetas disponibles. 10. Da clic en guardar cambios.		11. Elimina de la base de datos, todos los registros que asocien al prodigio con alguna etiqueta. Si ocurre un error, ver Curso de Excepción 3. 12. Almacena en la base de datos un registro que asocie al prodigio con una etiqueta. Uno por cada etiqueta que se encontrase en la tabla de etiquetas asociadas. Si ocurre un error, ver Curso de Excepción 4. 13. Muestra una alerta con el mensaje “La lista de etiquetas se ha actualizado”. 14. Redirige el visualizador web a la pantalla de Curar Prodigio.
Curso Alterno 1		

Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo obtener la lista de etiquetas disponibles”. 2. Redirige el visualizador web a la pantalla de Curar Prodigio.
Curso de Excepción 2		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo obtener la lista de etiquetas asociadas”. 2. Redirige el visualizador web a la pantalla de Curar Prodigio.
Curso de Excepción 3		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo actualizar la lista de etiquetas del prodigio, inténtelo más tarde”.
Curso de Excepción 4		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo actualizar la lista de etiquetas del prodigio, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Establecer Presentación	
Objetivo	Permitir al curador establecer una disposición de los elementos del prodigio para su presentación dentro de la plataforma.	
Actor Principal	Curador	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Curar Prodigio.	
Postcondiciones	Un nuevo registro LayoutProdigio asociado con el prodigio curado.	
Flujo Básico		
Actor	Sistema	
1. Desde la pantalla de Curar Prodigio, da clic en el botón “Ajustar Presentación”.	2. Redirige el visualizador web a la pantalla de Establecer Presentación. 3. Busca en la base de datos la información del prodigio sobre el que se está trabajando. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla las disposiciones posibles que se pueden elegir, así como el formulario de presentación (Tipo de fuente para título, Tamaño de Fuente para título, Tipo de fuente para texto, Tamaño de fuente para texto, División de párrafos [Oculto]). En el lado derecho, muestra el área de selección de división de párrafo. En la parte inferior muestra los botones “Volver a Detalles del Prodigio”, “Ver vista Previa”, “Guardar Cambios”. Si el prodigio ya posee un objeto LayoutProdigio asociado ver Curso Alterno 1.	
5. Selecciona una de las disposiciones. Si la opción seleccionada no es la número 4, salta al paso 11 del Flujo Normal.	6. Habilita las opciones de separación de párrafo.	
7. Selecciona del área de texto de división de párrafos, la porción de texto que será separada. 8. Da clic en la opción “Mover Abajo”, indicada por una flecha hacia abajo.	9. Mueve el texto seleccionado al área de texto inferior. 10. Almacena en el campo de División de párrafo la posición del cursor en la que se separó el texto.	
11. Selecciona los tipos y tamaños de fuente para título y texto. 12. Da clic en el botón Guardar Cambios.	13. Almacena en la base de datos un nuevo registro LayoutProdigio con la información capturada en el formulario asociado al prodigio trabajado. Si el prodigio trabajado ya poseía un registro LayoutProdigio asociado, lo actualiza. Si ocurre un error, ver Curso de Excepción 3.	

	14. Muestra una alerta con el mensaje “Se ha guardado la presentación del prodigio”. 15. Redirige el visualizador web a la pantalla de Vista previa del prodigio.
16. Si dese volver a ajustar la presentación, ver Curso Alterno 2. 17. Da clic en el botón “Volver a Detalles del Prodigio”.	18. Redirige el visualizador web a la pantalla de Curar Prodigio.
Curso Alterno 1	
Actor	Sistema
	1. Carga en el Formulario la información del objeto LayoutProdigio asociado al prodigio. 2. Si el layout del objeto es 4, vuelve al paso 6 del Flujo Normal.
Curso Alterno 2	
Actor	Sistema
1. Da clic en el botón “Volver a Ajustar Presentación”.	2. Vuelve al punto 2 del Flujo Normal.
Curso de Excepción 1	
Actor	Sistema
	1. Muestra una alerta con el mensaje “No se pudo obtener la información del prodigio”. 2. Redirige el visualizador web a la pantalla de Curar Prodigio.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra una alerta con el mensaje “No se pudo guardar la presentación del prodigio, intente más tarde”.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Publicar Prodigio	
Objetivo	Permitir al Curador marcar un Prodigio específico como listo para publicar.	
Actor Principal	Curador.	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Curar Prodigio.	
Postcondiciones	El objeto Prodigio actualizado con estatus Publicado.	
Flujo Básico		
Actor		Sistema
1. Da clic en el botón “Publicar Prodigio”.		2. Actualiza el estatus del prodigio trabajado a “Publicado”. Si ocurre un error, ver Curso de Excepción 1. 3. Redirige el visualizador web a la pantalla de Listar Prodigios para Curador.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		Muestra una alerta con el mensaje “No se pudo publicar el prodigio, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Crear Convocatoria	
Objetivo	Permitir al Director registrar una nueva convocatoria en el sistema.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Convocatorias (Como Director)	
Postcondiciones	Un nuevo registro de Convocatoria en la base de datos.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Convocatorias (Director) da clic en el botón “Crear Nueva Convocatoria”.		2. Redirige el visualizador web a la pantalla de Crear Convocatoria.
4. Captura la información requerida por el formulario.		3. Carga el formulario de Convocatoria.
5. Da clic en el botón Guardar.		6. Almacena en la base de datos un nuevo registro de convocatoria, con el estatus Nueva. Si la fecha de Inicio registrada es igual a la fecha actual, el estatus por defecto será Abierta. Si ocurre un error, ver Curso de Excepción 1.
		7. Muestra una alerta con el mensaje “Convocatoria Guardada”.
		8. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo guardar la Convocatoria, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Listar Miembros del Consejo	
Objetivo	Permitir al Director enlistar los miembros del Consejo que participan en una convocatoria Específica.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Ver Detalle Convocatoria.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Da clic en el botón “Gestionar Consejo” en la pantalla de Ver Detalle de Convocatoria.		2. Redirige el visualizador web a la pantalla de Listar Miembros del Consejo. 3. Busca en la base de datos todos los Usuarios que poseen algún perfil dentro de la convocatoria seleccionada. Si ocurre un error, ir al Curso de Excepción 1. 4. Muestra en pantalla, dentro de una tabla, la lista de usuarios encontrados junto con los roles que desempeñan. En cada uno, agrega la opción “Ver Detalles”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “no se pudo recuperar la lista de Miembros del Consejo”. 2. Redirige el visualizador web a la pantalla de Ver Detalle Convocatoria.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Registrar Miembro del Consejo	
Objetivo	Permitir al Director registrar un nuevo miembro del consejo en una convocatoria específica.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Miembros del Consejo.	
Postcondiciones	Un nuevo registro de MiembroConsejo y, si es el caso, un nuevo registro de Usuario.	
Flujo Básico		
Actor		Sistema
1. Da clic en el botón “Registrar nuevo Miembro”, en la pantalla de Listar Miembros del Consejo.		2. Redirige el visualizador web a la pantalla de Buscar Nuevo Miembro del Consejo. 3. Carga el formulario de búsqueda de Usuarios (¿Correo, Nombre, Primer Apellido, Segundo Apellido, Ya ha participado?).
4. Si el Director desea registrar un nuevo miembro que no posee una cuenta en la plataforma, ver Curso Alterno 1. 5. Captura la información del usuario que desea buscar. 6. Da clic en el botón Buscar.		7. Busca en la base de datos todos los usuarios que coinciden con la información capturada en el formulario. Si ocurre un error, ir al Curso de Excepción 1. 8. Muestra en pantalla una lista con los usuarios encontrados, cada uno con la opción “Registrar Como Miembro”.
9. Da clic en la opción “Registrar Como Miembro” de uno de los usuarios encontrados.		10. Redirige el visualizador web a la pantalla de Registrar Miembro de Consejo. 11. Carga el formulario de registro de Miembro (Nombre, Primer Apellido, Correo, Contraseña, Inicio de Labores, Final de Labores, Rol) con los campos deshabilitados a excepción de Inicio de Labores, Final de Labores y Rol, con la información del usuario seleccionado.
12. Captura la información sobre su posición como miembro de consejo. 13. Da clic en el botón “Registrar”.		14. Guarda en la base de datos un nuevo registro MiembroConsejo con la información capturada por el Director. Si ocurre un error ver curso de Excepción 2. 15. Muestra una alerta con el mensaje “Se ha registrado al usuario seleccionado como miembro del consejo”. 16. Redirige el visualizador a la pantalla de Listar Miembros del Consejo.
Curso Alterno 1		

Actor		Sistema	
1. Da clic en el botón Registrar un nuevo Miembro.		2. Redirige el visualizador web a la pantalla de Registrar Miembro de Consejo. 3. Carga el formulario de registro de Miembro (Nombre, Primer Apellido, Correo, Contraseña, Inicio de Labores, Final de Labores, Rol).	
4. Captura la información requerida del usuario y sus roles como miembro del consejo. 5. Da clic en el botón “Registrar”.		6. Guarda en la base de datos un nuevo Registro de Usuario con la información capturada en el formulario de usuario. Si ocurre un error, ir al Curso de Excepción 3. 7. Guarda en la base de datos un nuevo registro de MiembroConsejo con la información capturada en el formulario de Roles. Si ocurre un error, ir al Curso de Excepción 2. 8. Muestra una alerta con el mensaje “Se ha registrado el nuevo Miembro del Consejo”. 9. Redirige el visualizador web a la pantalla de Listar Miembros del Consejo.	
Curso de Excepción 1			
Actor		Sistema	
		1. Muestra una alerta con el mensaje “No se pudo recuperar la lista de usuarios”.	
Curso de Excepción 2			
Actor		Sistema	
		1. Muestra una alerta con el mensaje “Ocurrió un error al intentar registrar el miembro del Consejo, inténtelo más tarde”. 2. Redirige a la pantalla de Buscar Nuevo Miembro del Consejo.	
Curso de Excepción 3			
Actor		Sistema	
		1. Muestra una alerta con el mensaje “No se pudo registrar el usuario, inténtelo mpas tarde”. 2. Redirige a la pantalla de Buscar Nuevo Miembro del Consejo.	
Requerimientos no funcionales específicos		Ninguno	

Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Ver Detalle Miembro del Consejo	
Objetivo	Permitir al director ver la información de un miembro del consejo en específico.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Miembros del Consejo.	
Postcondiciones	Ninguna	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Miembros del Consejo, dar clic en la opción “Ver Detalles” de alguno de los usuarios listados.		2. Redirige el visualizador web a la pantalla de Ver Detalle Miembro del Consejo. 3. Busca en la base de datos la información del usuario seleccionado con su perfil en la convocatoria. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla la información recuperada de la base de datos. En el caso de la información de Miembro de Consejo (Inicio de Labores, Final de Labores, Rol), muéstrala en un formulario habilitado. En la parte inferior muestra los botones “Actualizar” y “Dar de Baja”.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo recuperar la información del Miembro del Consejo”. 2. Redirige el visualizador web a la pantalla de Listar Miembros del Consejo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Eliminar Miembro del Consejo	
Objetivo	Permitir al Director dar de baja a un Miembro del Consejo de una convocatoria específica.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso “Ver detalle Miembro del Consejo”.	
Postcondiciones	El registro MiembroConsejo eliminado de la base de datos.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla “Ver Detalle Miembro del Consejo” da clic en el botón “Dar de Baja”.		2. Elimina de la base de datos el registro MiembroConsejo que coincida con el usuario seleccionado y la convocatoria en la que participa. Si ocurre un error, ver Curso de Excepción 1. 3. Muestra una alerta con el mensaje “Se ha dado de baja a este Miembro del Consejo”. 4. Redirige el visualizador a la pantalla de Listar Miembros del Consejo.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo dar de baja a este Miembro del Consejo, inténtelo más tarde”. 2. Redirige el visualizador a la pantalla de Listar Miembros del Consejo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Actualizar Miembro del Consejo	
Objetivo	Permitir al Director actualizar la información de Miembro de Consejo de un usuario que participe en una de sus convocatorias.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Ver Detalle Miembro de Consejo.	
Postcondiciones	El registro de Miembro de Consejo del Usuario seleccionado actualizado.	
Flujo Básico		
Actor		Sistema
1. En la pantalla de Ver Detalle Miembro del Consejo, captura la nueva información del Miembro de Consejo (Inicio de Labores, Final de Labores, Rol).		3. Actualiza en la base de datos la información del registro MiembroConsejo que coincida con la Convocatoria y el Usuario seleccionados. Si ocurre un error, ver Curso de Excepción 1.
2. Da clic en el botón “Actualizar”.		4. Muestra una alerta con el texto “La información del Miembro de Consejo se ha actualizado”.
		5. Redirige el visualizador web a la pantalla de Listar Miembros del Consejo.
Curso Alterno 1		
Actor		Sistema
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el texto “No se puedo actualizar la información del Miembro del Consejo, inténtelo más tarde”.
		2. Redirige el visualizador web a la pantalla de Listar Miembros del Consejo.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Evaluar Prodigios	
Objetivo	Permitir al director ver las calificaciones realizadas sobre los prodigios, por parte de los Revisores, subidos de la convocatoria y en base a lo observado decidir si Aprobar, Rechazar o solicitar la Corrección del Prodigio.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Convocatorias.	
Postcondiciones	Un registro prodigio actualizado con estatus: Aceptado, Rechazado o En Corrección.	
Flujo Básico		
Actor		Sistema
1. Desde la pantalla de Listar Convocatorias, dar clic en la opción “Evaluar Prodigios”.		2. Redirige el visualizador web a la pantalla de Evaluar Prodigios. 3. Busca en la base de datos todos los Prodigios que ya hayan sido evaluados, por lo menos una vez, por alguno de los revisores. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en la pantalla la lista de Prodigios obtenida, cada uno con el número de votos de Aceptación, Rechazo y Solicitud de Corrección, junto con 3 botonees de acción, Aceptar, Rechazar y Solicitar Corrección. Si el prodigio se encuentra con estatus “Aprobado”, “Rechazado” o “Por corregir”, deshabilita los botones de acción. En la parte inferior de la lista, muestra el botón “Abrir Galería”, por defecto deshabilitado o si la galería se encuentra con estatus “Galería Abierta”. 5. Comprueba el estatus de todos los prodigios listados. Si todos los prodigios listados no se encuentran con estatus “Nuevo”, “En revisión” o “Por Corregir”, habilita el botón “Abrir Galería”.
6. Si el Director desea Solicitar la Corrección de un Prodigio, Ver Curso Alterno 1. 7. Si el Director desea Rechazar un Prodigio, ver Curso Alterno 2.		9. Actualiza en la base de datos el estatus del prodigio seleccionado a “Aceptado”. Si ocurre un error, ver curso de Excepción 2. 10. Muestra una alerta con el mensaje “Se ha Aprobado el Prodigio Seleccionado”.

8. Si el Director desea Aceptar un Prodigio, da clic en el botón “Aceptar” de uno de los prodigios listados.	11. Deshabilita los botones de acción del prodigio seleccionado. 12. Comprueba el estatus de todos los prodigios listados. Si todos los prodigios listados no se encuentran con estatus “Nuevo”, “En revisión” o “Por Corregir”, habilita el botón “Abrir Galería”.
Curso Alterno 1	
Actor	Sistema
1. Da clic en el botón “Solicitar Corrección”.	2. Actualiza en la base de datos el estatus del prodigio seleccionado a “Por Corregir”. Si ocurre un error, ver curso de Excepción 2. 3. Muestra una alerta con el mensaje “Se ha Aprobado el Prodigio Seleccionado”. 4. Deshabilita los botones de acción del prodigio seleccionado. 5. Comprueba el estatus de todos los prodigios listados. Si todos los prodigios listados no se encuentran con estatus “Nuevo”, “En revisión” o “Por Corregir”, habilita el botón “Abrir Galería”.
Curso Alterno 2	
Actor	Sistema
1. Da clic en el botón “Rechazar”.	2. Actualiza en la base de datos el estatus del prodigio seleccionado a “Rechazado”. Si ocurre un error, ver curso de Excepción 2. 3. Muestra una alerta con el mensaje “Se ha Rechazado el Prodigio Seleccionado”. 4. Deshabilita los botones de acción del prodigio seleccionado. 5. Comprueba el estatus de todos los prodigios listados. Si todos los prodigios listados no se encuentran con estatus “Nuevo”, “En revisión” o “Por Corregir”, habilita el botón “Abrir Galería”.
Curso de Excepción 1	
Actor	Sistema

		1. Muestra una alerta con el mensaje “No se pudo recuperar el listado de prodigios”. 2. Redirecciona al panel de director.
Curso de Excepción 2		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo tomar acción sobre el prodigio”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

Caso de Uso	Abrir Galería	
Objetivo	Permitir al Director abrir la visualización de los prodigios evaluados al público.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Evaluar Prodigios y haber evaluado todos los prodigios listados.	
Postcondiciones	Estatus de la “Convocatoria” actualizado a “Galería Abierta”.	
Flujo Básico		
Actor		Sistema
1. Da clic en el botón Abrir Galería.		2. Actualiza el estatus de la convocatoria sobre la que se trabaja a “Galería Abierta”. Si ocurre un error, ver Curso de Excepción 1 3. Muestra una alerta con el mensaje “Galería abierta con éxito”. 4. Redirige el visualizador web a la pantalla de Listar Convocatorias.
Curso Alterno 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo abrir la galería, inténtelo más tarde”.
Curso de Excepción 1		
Actor		Sistema
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	
Caso de Uso	Listar Comentarios Reportados	
Objetivo	Permitir al Director visualizar la lista de comentarios reportados encontrados en prodigios de sus convocatorias.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma.	
Postcondiciones	Ninguna.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Comentarios Reportados” del menú lateral.		2. Redirige el visualizador web a la pantalla de Listar Comentarios Reportados. 3. Busca en la base de datos todos los comentarios que poseen uno o más reportes que se encuentren en prodigios de convocatorias del Director en

	Sesión. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla los prodigios encontrados junto a la cantidad de reportes que posee. En cada elemento, ofrece la opción “Ver Detalles”.
Curso Alterno 1	
Actor	Sistema
Curso de Excepción 1	
Actor	Sistema
	1. Muestra una alerta con el mensaje “No se pudo obtener la lista de comentario, inténtelo más tarde”. 2. Redirige el visualizador web a la pantalla de inicio del Panel de Director.
Requerimientos no funcionales específicos	Ninguno
Pendientes	Ninguno
Supuestos	Ninguno

Caso de Uso	Atender Comentario Reportado	
Objetivo	Permitir al Director tomar acción sobre los comentarios reportados.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma, Haber ejecutado el caso de uso Listar Comentarios Reportados.	
Postcondiciones	Actualización del estatus del comentario seleccionado. Si es el caso, un nuevo registro de Notificación o un nuevo registro de SolicitudBajausuario.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Ver Detalles” de alguno de los comentarios listados.		2. Redirige el visualizador web a la pantalla de Atender Comentario Reportado. 3. Busca en la base de datos el Comentario seleccionado, así como los reportes asociados a este y el Participante que lo comentó. Si ocurre un error, ver Curso de Excepción 1. 4. Muestra en pantalla los detalles del comentario seleccionado y la lista de reportes encontrados. 5. En la parte inferior, carga el formulario de acciones a tomar (Enviar advertencia o Solicitar baja de usuario) junto con los botones “Bloquear comentario” y “Reactivar Comentario”. Este último botón aparece como deshabilitado si el comentario se encuentra con estatus Activo, en el caso contrario se encuentra Deshabilitado.
6. Si el comentario se encuentra inactivo, ver Curso Alterno 1. 7. Si desea solicitar la baja del Participante, ver Curso Alterno 2. 8. Marca la opción “Enviar Advertencia” del formulario de acciones a tomar. 9. Dar Clic en el Botón “Bloquear Comentario”.		10. Actualiza en la base de datos el estatus del comentario seleccionado a “Inactivo”. Si ocurre un error, ver Curso de Excepción 1. 11. Almacena en la base de datos un nuevo registro Notificación, con remitente el Director en sesión, como Destinatario el Participante que realizó el comentario y como Asunto la Advertencia de Reincidencia. Si ocurre un error, ver Curso de Excepción 2. 12. Muestra una alerta con el mensaje “Acción realizada exitosamente”.

	13. Redirige el visualizador web a la pantalla de Listar Comentarios Reportados.
Curso Alterno 1	
Actor	Sistema
1. Da clic en el botón “Reactivar Comentario”.	2. Actualiza en la base de datos el estatus del comentario seleccionado a “Inactivo”. Si ocurre un error, ver Curso de Excepción 1. 3. Muestra una alerta con el mensaje “Acción realizada exitosamente”. 4. Redirige el visualizador web a la pantalla de Listar Comentarios Reportados.
Curso Alterno 2	
Actor	Sistema
1. Marca la opción “Solicitar Baja de usuario” del formulario de acciones a tomar. 2. Dar Clic en el Botón “Bloquear Comentario”.	3. Actualiza en la base de datos el estatus del comentario seleccionado a “Inactivo”. Si ocurre un error, ver Curso de Excepción 1. 4. Almacena en la base de datos un nuevo registro SolicitudBajaUsuario, teniendo como solicitante al Director en sesión y como Afectado el Participante que realizó el comentario. Si ocurre un error, ver Curso de Excepción 3. 5. Muestra una alerta con el mensaje “Acción realizada exitosamente”. 6. Redirige el visualizador web a la pantalla de Listar Comentarios Reportados.
Curso de Excepción 1	
Actor	Sistema
	1. Muestra una alerta con el mensaje “La acción no pudo ser realizada, inténtelo más tarde”.
Curso de Excepción 2	
Actor	Sistema
	1. Muestra una alerta con el mensaje “La acción no pudo ser realizada, inténtelo más tarde”.
Curso de Excepción 3	
Actor	Sistema

		1. Muestra una alerta con el mensaje “La acción no pudo ser realizada, inténtelo más tarde”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	

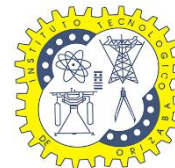
Caso de Uso	Atender Solicitudes de Baja de Prodigios	
Objetivo	Permitir al Director atender las Solicitudes de Bajas de Prodigio realizadas sobre los Prodigios subidos en sus convocatorias.	
Actor Principal	Director	
Precondiciones	Haber iniciado sesión en la plataforma.	
Postcondiciones	Un registro de prodigio con estatus actualizado.	
Flujo Básico		
Actor		Sistema
1. Da clic en la opción “Solicitudes de baja de Prodigios” del menú lateral.		2. Redirige el visualizador web a la pantalla de Atender Solicitudes de baja de Prodigio. 3. Busca en la base de datos todas solicitudes de baja de prodigios que aún no hayan sido atendidas y que pertenezcan a convocatorias del Director en sesión. Si ocurre un error, ver curso de Excepción 1. 4. Muestra la lista de prodigios encontrados, cada uno con las opciones “Dar de Baja” y “Mantener”.
5. Si desea Mantener el prodigio e ignorar la solicitud, ver Curso Alterno 1. 6. Da clic en la opción “Dar de Baja”.		7. Actualiza en la base de datos el estatus del prodigio indicado en la solicitud a “Dado de Baja”. Si ocurre un error, ver curso de Excepción 2. 8. Muestra una alerta con el mensaje “Acción realizada exitosamente”.
Curso Alterno 1		
Actor		Sistema
1. Da clic en el botón Mantener		
Curso de Excepción 1		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo obtener la lista de Solicitudes de Baja de Prodigio”. 2. Redirecciona el visualizador web a la pantalla de inicio del panel del Director.
Curso de Excepción 2		
Actor		Sistema
		1. Muestra una alerta con el mensaje “No se pudo realizar la acción”.
Requerimientos no funcionales específicos	Ninguno	
Pendientes	Ninguno	
Supuestos	Ninguno	



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Tecnológico Nacional de México

Instituto Tecnológico de Orizaba

División de Estudios de Posgrado e Investigación

Maestría en Sistemas Computacionales

TÍTULO:

Servicios de la Plataforma

PRESENTADO POR:

I.S.C. Adolfo Meza Romero M15011191

Índice

Servicios de la Plataforma	1
---	----------

Servicios de la Plataforma

Caso de Uso	ID	Recurso	Método	URL (app/apiprod/)	Parámetro GET/DELETE -> Query String POST/PUT -> JSON	JSON Salida (Formato de Fecha AAAA/MM/DD)	Código HTTP		Notas: Se devuelve código 400 en cualquier servicio que genere error por datos.	DTO/Serializer Necesarios Nota, cuando es un objeto completo, ya existe el serializador (Modelo)	Proceso almacenado
Ver Galería	1	prodigio	GET	prodigios/mapa	nombre, municipio, localidad, cveCat[](como php), tipoMIME[]	[{ id:number, lat:number long:number nombre:String categoria:{ id:String, descrip:String, icono:String, sombra:String } }]	200	Ok		Serializador de entrada: ProdigioMapaParamSerializer Dado que algunos campos no se emplean, es posible colocarlos como required=False	
							404	Not Found		Serializador de Salida ProdigioMapaReturnSerializer CategoriaMapaReturnSerializer	
	2	prodigio	GET	prodigios/galeria	tipoMIME[], cveEtiqueta[], nombre, cveCat, cveConvocatoria(opcional)	[{ id:num, nombre:String, imgMin:String }]	200	Ok		Serializador de entrada ProdigioGaleriaParamSerializer Dado que algunos campos no se emplean, es posible colocarlos como required=False	Se requiere proceso para convertir de binario a base 64
							404	Not Found		Serializador de Salida ProdigioGaleriaReturnSerializer	
	30	etiqueta	GET	etiquetas/por_categoria/<int:idCat>	idCategoria (En la ruta)	[{ Arreglo de Objetos Etiqueta }]	200	Ok	Si el id de la categoría es 0, obtén todas.		
							404	Not Found			
	3	convocatoria	GET	convocatorias/galería/recientes		[{ id:num,	200	Ok	Rango de búsqueda por definir		

						titulo:String, poster:String }}	404	Not Found	Busca únicamente las convocatorias con estatus Galería Abierta	Serializador de salida ConvocatoriaGaleriaReturnSerializer	Se requiere proceso para convertir de binario a base 64
	4	convocatoria	GET	convocatorias/por_filtro	idDirector idRevisor idCurador galería: boolean	Arreglo de Convocatoria	200	Ok	Si los 3 parámetros van vacíos, trae todas sin condiciones. Si el id es de director trae todas las convocatorias registradas por el director Si el id es de revisor, trae las convocatorias en las que participa como Revisor Si el id es de curador, trae las convocatorias en la que participa como Curador Si es galería, busca únicamente las convocatorias con estatus Galería Abierta	Serializador de entrada ConvocatoriaFiltroParamSerializer Dado que algunos campos no se emplean, es posible colocarlos como required=False	
							404	Not Found			
Ver Prodigio	5	prodigio	GET	prodigios/<int:id>	Id (en la ruta)	Objeto Prodigio	200	Ok	Cuando un revisor vea un prodigio de video, con un diferencial de subida y hora actual menor a 15 min, indica el video aún se está procesando.		Se requiere proceso para convertir de binario a base 64
							404	Not Found			

	6	comentario	GET	comentarios/por_prodigio	idProdigio, activo:booleano enRevision:boolean bloqueado:boolean	[[id:num, descript:String fechaPub:String estatus:number autor:{ todo el objeto }]]	200	Ok		ComentarioPorProdigioParamSerializer	
							404	Not Found		ComentarioPorProdigioReturnSerializer	
	71								Este trae todas las reacciones, con sus respectivos objetos.		
	75	reaccion	GET	reacciones/por_prodigio/<int:idProd>	idProdigio	[[Cantidad de reacciones + Objeto Reaccion]]	200, 404			Retorno ReaccionesPorProdigioRetornSerializer	
	76	reaccion	POST	reacciones/por_prodigio	idProdigio idReaccion	{Cantidad de reacciones + Objeto Reaccion}	201, 500, 404		Si no hay registrada ninguna reacción de ese tipo para ese prodigio, crea un registro nuevo ReaccionProdigio con cant 1. De otro modo, se actualiza aumentando en 1 la cantidad. En el LocalStorage se guardan los identificadores de los prodigios calificados, de modo que no se	Entrada ReaccionesPorProdigioParamSerializer Retorno ReaccionesPorProdigioRetornSerializer	

									permite calificar 2 veces el mismo prodigio.		
Reportar Comentario	7	reporte	POST	reportes	idComentario descripcion	[id:num descrip:String]	201	Created		Entrada ReporteComentarioParamSerializer	
							409	No existe el comentario		Retorno ReporteComentarioRetornSerializer	
							500	Error interno			
Autoregistrarse en la plataforma	8	usuario	POST	usuarios	correo contrasena nombre prmAll segAppl	[Objeto Usuario]	201	Created		Entrada UsuarioRegistroParamSerializer	Cifrar contraseña (MD5)
							500	Error interno			
Ingresar a la Plataforma	9	token	GET	token/iniciar_sesion	correo contrasena	{ Token: String, esDirector: boolean esRevisor: boolean esCurador: boolean esAdmin: boolean }	200	Ok		Entrada InicioSesionParamSerializer	Cifrar contraseña (MD5)
							404	Not Found		Retorno InicioSesionReturnSerializer	
Cerrar Sesión	23	token	DELETE	token/cerrar_sesion			204		Requiere del token en la Cabecera		
Solicitar Baja de Prodigio	10	solicitudBajaP	POST	solicitudes_baja_p	fechaSol motivo idProdigio correoSolicitante	[Objeto SolicitudBaja]	201	Created	Requiere del token en la Cabecera	Entrada SolicitudBajaParamsSerializer	
							500	Error interno			
Reportar Prodigio	10										

Subir Prodigio	11	categoria	GET	categorias		[Arreglo de categorías]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
	12	estado	GET	estados		[Arreglo de estados]	200	Ok			
							404	Not Found			
	13	municipio	GET	municipios/por_estado/<int:idEst>	clvEstado (En la ruta)	[Arreglo de minicipios]	200	Ok			
							404	Not Found			
	14	localidad	GET	localidades/por_municipio/<int:idMun>	clvMunicipio (En la ruta)	[Arreglo de localidades]	200	Ok			
							404	Not Found			
	15	Código postal	GET	codposts/por_localidad/<str:idLoc>	clvLocalidad (En la ruta)	[Arreglo de Códigos postales]	200	Ok			
							404	Not Found			
	16	prodigio	POST	prodigios	titulo(nombreArch) fechaCaptura tipoArch contArch lat long ubicado descripcion codPost categoria	{ Objeto Prodigio }	201	Created	Requiere token en la cabecera	Entrada ProdigioParamsSerializer	Proceso almacenado para convertir de base64 a binario
							500	Error interno			

Dejar Comentario	17	comentario	POST	comentarios/por_prodigio	descripcion idUsuario idProdigio	{ Objeto Comentario }	201	Created	Requiere token en la cabecera	Entrada ComentarioProdigioParams	
							500	Error interno			
Listar Prodigios Subidos	18	prodigio	GET	prodigios/por_usuario/<str:idUs>	idUsuario (En la ruta)	[Arreglo de Objetos Prodigio]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
Ver Detalles Prodigio	5										
Listar Convocatorias	4										
Corregir Prodigio	5										
	20	revision	GET	revisiones/por_prodigio/<int:idProd>	idProdigio (En la ruta)	[Arreglo de Objetos Observación]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
	21	prodigio	PATCH	prodigios/correccion	id titulo(nombreArch) descripción categoría	{ Objeto Prodigio }	200	Ok	-Requiere token en la cabecera -Generar notificación de corrección desde el servicio	Entrada ProdigioCorreccionParamsSerializer	Se requiere proceso para convertir de binario a base 64 (Retorno)
							500	Error interno			
Listar Notificaciones	24	notificacion	GET	notificaciones/por_destino/<str:idUs>	idUsuario (En la ruta)	[Arreglo de Objetos Notificacion]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
Ver Detalles Notificación	25	notificacion	GET	Notificaciones/<int:id>	Id (En la ruta)	{ Objeto Notificacion }	200	Ok	Requiere token en la cabecera		
							404	Not Found			
Ver Detalle Convocatoria	26	convocatoria	GET	Convocatorias/<int:id>	Id (En la ruta)	{ Objeto	200	Ok			

						Convocatoria }	404	Not Found			Se requiere proceso para convertir de binario a base 64
Listar Prodigios para Revisor	27	prodigio	GET	prodigios/por_panel	idPanel idConvocatoria	[Arreglo de Objetos Prodigio (id, nombreArch, fechaSub, estatus, fechaCorr, cantRev)]	200	Ok	-Requiere token en la cabecera -cantRev se genera haciendo Count(Revisiones)	Entrada ProdigioPorPanelParamsSerializer	
							404	Not Found		Salida ProdigioPorPanelRetornSerializer	
Revisar Prodigio	5										
	20										
	28	revision	POST	revisiones	idProdigio idUsuario observación calificación correccion	{ Objeto Revision }	201	Created	Requiere token en la cabecera	Entrada RevisionesParamsSerializer	
							500	Error interno			
Listar Prodigios para Curador	27										
Curar Prodigio	5										
	29	prodigio	PATCH	prodigios/reubicacion	Id lat long	{ Objeto Prodigio }	200	Ok	Requiere token en la cabecera	Entrada ProdigioReubicacionParamsSerializer	Se requiere proceso para convertir de binario a base 64 (Retorno)
							500	Error interno			
							404	Not Found			
	30										
Agregar Etiquetas	31	etiqueta	GET	etiquetas/por_prodigio/<int:idProd>	idProdigio (En la ruta)	[Objeto Etiqueta]	200	Ok	Requiere Token en la cabecera -El 404 en el cliente no genera error, se espera si		
							404	Not Found			

									es que no tiene etiquetas asignadas.		
	32	prodigio	PATCH	prodigios/etiquetas	idProdigio [idEtiquetas]	{ Objeto Prodigio }	200	Ok	-Requiere Token en la Cabecera. -El objeto prodigio incluye un arreglo de etiquetas. -Internamente, primero se borran las etiquetas asociadas y después se insertan las nuevas. -Se llama un proceso almacenado	Entrada ProdigiosEtiquetasActualizParamSerializer	
							500	Error interno			
							404	Not Found			
Establecer Presentación	5										
	33	prodigio	PATCH	prodigios/layout	id layout fuenteTiutulo tamanoTitulo fuenteTexto tamanoTexto separacion	{ Objeto Prodigio }	200	Ok	Requiere token en la cabecera	Entrada ProdigioLayoutParamsSerializer	
							500	Error interno			
							404	Not Found			
Publicar Prodigio	34	prodigio	PATCH	prodigios/publicación/<int:idProd>	Id (En la ruta)	{ Objeto Prodigio }	200	Ok	Requiere token en la cabecera		
							500	Error interno			
							404	Not Found			
	36	miembroConsejo	GET	miembros_consejo/por_convocatoria/		[{	200	Ok			

Listar Miembros del Consejo				<int:idConv>	idConvocatoria (En la ruta)	Objeto Miembro Consejo }}	500	Error interno	Requiere Token en la cabecera		
							404	Not Found			
Registrar Miembro del Consejo	37	usuario	GET	usuario/por_filtro	nombre prmAppl segAppl correo soloMiembros	[[Objetos Usuario]]	200	Ok	Requiere Token en la Cabecera	Entrada UsuarioFiltroParamsSerializer	
							500	Error interno			
							404	Not Found			
	38	usuario	GET	usuarios/<int:id>	Id (En la ruta)	{ Objeto Usuario }	200	Ok	Requiere Token en la cabecera Solo existe si no hay forma de pasar datos complejos a un componente (property Value)		
							500	Error interno			
							404	Not Found			
	39	miembroConsejo	POST	miembros_consejo	Perfil[] iniLabores finLabores idConvocatoria idUsuario	{ Objeto Miembro de Consejo }	201	Created	Requiere Token en la Cabecera	Entrada MiembroConsejoParamsSerializer	Verificar inserción, modificación de relaciones 1 a muchos en django
							500	Error interno	Primero borra relación usuario-convocatoria, luego inserta		
	40	usuario	POST	usuarios/miembro_consejo	correo contrasena nombre prmAppl segAppl	{ Objeto usuario + Miembro Consejo	201	Created	Requiere Token en la Cabecera	Entrada UsuarioMiembroParamsSerializer	Verificar inserción, modificación de relaciones 1 a muchos en django

					Perfil[] iniLabores finLabores idConvocatoria	}	500	Error interno			
Ver Detalle Miembro del Consejo	41	miembroConsejo	GET	miembros_consejo	idUsuario idConvocatoria	{ Objeto Miembro de Consejo }	200	Ok	Requiere Token en la Cabecera	Entrada MiembroConsejoDetailsParamsSerializer	
							500	Error interno			
							404	Not Found			
Eliminar Miembro del Consejo	42	miembroConsejo	DELETE	miembros_consejo	idUsuario idConvocatoria		204		Requiere Token en la Cabecera Se actualiza la fecha de fin de labores.	Entrada MiembroConsejoElimnParamsSerializer	
Actualizar Miembro del Consejo	39										
Crear Convocatoria	43	convocatoira	POST	convocatorias	Idusuario titulo fechaAp fechaFn bases	{ Objeto Convocatoria }	201	Created	Requiere token en la cabecera Si la fecha actual es distinta a FechaAp, entonces	Entrada ConvocatoriaParamsSerializer	Convertir base 64 a binario

					Poster		500	Error interno	es nueva. De lo contrario es Abierta		
Evaluar Prodigios	44	prodigio	GET	prodigios/calificados/<int:idConv>	idConvocatoria (En la ruta)	[[Objetos Convocatoria]]	200	Ok	Requiere token en la cabecera Son todos los prodigios de la convocatoria que sean nuevos, en revisión o por corregir		
							500	Error interno			
							404	Not Found			
	45	prodigio	PATCH	prodigios/evaluacion	Estatus idProdigio	{ Objeto Prodigio }	200	Ok	Requiere token en la cabecera (Aceptado, Rechazado, En correccion) Ademas de modificar el estatus del prodigio, genera una notificación para el participante	Entrada ProdigioEvaluacionParamsSerializer	
							500	Error interno			
Abrir Galeria	46	convocatoria	PATCH	convocatorias/<int:idConv>/abre_galeria	idConvocatoria (En la ruta)	{ Objeto Convocatoria }	200	Ok	Requiere token en la cabecera		
							500	Error interno			
	47	comentario	GET	comentarios/reportados/<int:idDir>	idDirector (En la ruta)	[[Objeto	200	Ok	Requiere token en la cabecera		

Listar Comentarios Reportados						Comentario + Cant Reportes {Obj Usuario} }]	500	Error interno			
							404	Not Found			
Atender Comentario Reportado	48	comentario	GET	comentarios/<int:id>	idComentario (En la Ruta)	{ Objeto Comentario [{Obj Reporte}] {Obj Usuario} }	200	Ok	Requiere token en la cabecera		
							500	Error interno	Es posible que django devuelva el objeto prodigio asociado vacío		
							404	Not Found			
	49	comentario	PATCH	comentarios	idComentario Estatus Acción:integer (0=Omitir, 1=Advertir, 2=Bloquear)	{ Objeto Comentario }	200	Ok	Requiere token en la cabecera	Entrada ComentarioActualizacionParamsSerializer	
							500	Error interno	Aplica tanto para bloquear como para reactivar comentario Ademas de actualizar el comentario, si la acción es 1, enviara una notificación de advertencia al participante, si es 2, enviara solicitud de baja al administrador y si es 0 no realizará ninguna acción		

Atender Solicitudes de Baja de Prodigios	50	solicitudBajaP	GET	solicitudes_baja_p /por_director/<str:idDir>	idDirector (En la ruta)	[Objeto SolicitudBajaProdigos]	200	Ok	Requiere token en la cabecera		
							500	Error interno			
							404	Not Found			
	51	solicitudBajaP	PATCH	solicitudes_baja_p	idSolicitud estatus	{ SolicitudBajaProdigos }	200	Ok	Requiere token en la cabecera	Entrada SolicitudBajaEstatusSerializer	
							500	Error interno			
Gestionar directores	52	usuario	GET	usuarios/directores/todos		[Objeto usuario]	200	Ok	Requiere token en la cabecera Solo trae usuarios con perfil de director		
							404	Not Found			
	53	usuario	POST	usuarios/directores	Nombre primerApll segundoAll Correo Contrasena	{ Objeto usuario }	201	Created	Requiere token en la cabecera Por defecto da el rol de director al usuario	Entrada DirectorParamsSerializer	
							500	Error interno			
	38										
	54	usuario	PATCH	usuarios	IdUsuario Nombre primerApll segundoAll	{ Objeto usuario }	200	Ok	Requiere token en la cabecera	Entrada UsuarioActualizarParamsSerializer	
							500	Error interno			

					Correo Contrasena		404	Not Found			
Gestionar Categorías	11										
	55	categoria	POST	categorias	Descripción Icono Sombra Miniatura	{ Objeto Categoría }	201	Created	Requiere token en la cabecera	Entrada CategoriaParamsSerializer	Convertir base64 a binario
							500	Error interno			
	56	categoria	GET	categorias/<int:id>	Id (en la ruta)	{ Objeto Categoría }	200	Ok	Requiere token en la cabecera		Convertir de binario abase64
							404	Not Found			
	57	categoria	PATCH	categorias	Id Descripción Icono Sombra Miniatura	{ Objeto Categoría }	200	Ok	Requiere token en la cabecera	Entrada CategoriaActualizacionParamsSerializer	Convertir base64 a binario
							404	Not Found			
	58	categoria	PATCH	categorías/habilitado	Id Habilitado	{ Objeto Categoría }	200	Ok	Requiere token en la cabecera	Entrada CategoriaEstatusParamsSerializer	
							404	Not Found			
Gestionar Perfiles	59	perfil	GET	perfiles/todos		[{ Objeto Perfil }]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
	60	funcion	GET	funciones/todos		[{ Objeto Funcion }]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
	61	perfil	POST	perfiles	Nombre Funciones[]	{ Objeto Perfil }	201	Created	Requiere token en la cabecera	Entrada PerfilesParamsSerializer	
							500	Error interno			
	62	perfil	GET	Perfiles/<int:id>	Id (En la ruta)	{ Objeto Perfil + Funcion[] }	200	Ok	Requiere token en la cabecera		
							404	Not Found			

	63	perfil	PATCH	perfiles	Id Nombre Funcion[]	{ Objeto Perfil + Funcion[] }	200	Ok	Requiere token en la cabecera	Entrada PerfilActualizacionParamsSerializer	
							404	Error interno	Primero eliminará todas las funciones asociadas al perfil y posteriormente registrará las nuevas funciones recibidas.		
							500	Not Found			
Gestionar Funciones	60										
	64	funcion	POST	funciones	Nombre routerLink	{ Objeto Funcion }	201	Created	Requiere token en la cabecera	Nota: Se emplea el serialaizador estandar de Funcion, sin embargo, el id no es necesario en este caso	
							500	Error interno			
	79	funcion	GET	funciones/<int:idFunc>	Id (En la Ruta)	{ Objeto Funcion }	200	Ok	Requiere token en la cabecera		
							500	Error interno			
							404	Not Found			
	65	funcion	PATCH	funciones	Id Nombre RouterLink	{ Objeto Funcion }	200	Ok	Requiere token en la cabecera	Nota: Se emplea el serialaizador estandar de Funcion.	
							500	Error interno			
							404	Not Found			
Gestionar Etiquetas	30										
	66	etiqueta	POST	etiquetas	Descripcion IdCategoría	{ Objeto Etiqueta }	201	Created	Requiere token en la cabecera	Nota: Se emplea el serialaizador estandar de Etiqueta	
							500	Error interno			
81	etiqueta	GET	etiquetas/<int:idEtiqueta>	Id (En la Ruta)	{ Objeto Etiqueta }	200	Ok				

							500	Error interno	Requiere token en la cabecera		
							404	Not Found			
	67	etiqueta	PATCH	etiquetas	IdEtiqueta Nombre IdCategoría	{ Objeto Etiqueta }	200	Ok	Requiere token en la cabecera	Nota: Se emplea el serialaizador estandar de Etiqueta	
							500	Error interno	Solo modifica la etiqueta si no se ha usado		
404							Not Found				
Atender Solicitudes de Baja	68	solicitudBajaU	GET	solicitudes_baja_u/no_atendidas		[{Objeto SolicitudBajaUsuario}]	200	Ok	Requiere token en la cabecera		
							404	Not Found			
	69	solicitudBajaU	PATCH	solicitudes_baja_u/<int:idSol>	IdSolicitud (En la Ruta)	{Objeto SolicitudBajaUsuario}	200	Ok	Requiere token en la cabecera		
							500	Error interno			
404							Not Found				
Dar de Baja usuario	70	usuario	DELETE	usuarios/<str:idUsu>	Idusuario (En la Ruta)		200	Ok	Requiere token en la cabecera		
							404	Not Found	Cambia estatus a 0		
Gestionar Reacciones	71	reaccion	GET	reacciones/todas		[{Objeto Reaccion}]	200	Ok			
							404	Not Found			
	72	reaccion	POST	reacciones	descripcion icono	{Objeto Reaccion}	201	Created	Requiere Token en la cabecera	Nota: Se emplea el serialaizador estandar de Reaccion, es estatus se define explícitamente como true al crear el objeto	
							500	Error interno			
	73	reaccion	PATCH	reacciones	Id	{Objeto Reaccion}	200	Ok			

					descripcion icono		500	Error interno	Requiere token en la cabecera	Nota: Se emplea el serialaizador estandar de Reaccion, es estatus se define explícitamente como true al crear el objeto	
							404	Not Found			
	80	reaccion	GET	reacciones/<int:idReaccion>	Id (En la Ruta)	{Objeto Reaccion}	200	Ok	Requiere token en la cabecera		
							404	Not Found			
							500	Error interno			
	74	reaccion	DELETE	reacciones/<int:idReaccion>	Id (En la Ruta)		200	Ok	Requiere token en la cabecera		
							404	Not Found	Cambia habilitado a falso		
	77	usuario	PATCH	usuarios/contrasenia	idUserio contrasena	{Objeto Usuario}	200	Ok	Verifica que restCon = true. Actrualiza la contraseña y cambia restCon = false	Reutiliza Serializador de Inicio de sesion	Cifrar contraseña (MD5)
							500	Error interno			
							404	Not Found			
Solicitar Cambio de Contraseña	78	usuario	PATCH	usuarios/<str:isUsu>/rest_con	idUserio (En la Ruta)	{Objeto Usario}	200	Ok	Envia correo con url de restablecer contraseña. Actualiza usuario, resCon = true		
							404	Not Found			