

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN QUE PERMITA APOYAR
EL PROCESO DE SELECCIÓN DE BENEFICIARIOS DE PROYECTOS PÚBLICOS,
EJECUTADOS POR LA ALCALDÍA DEL MUNICIPIO DE TIMBÍO – CAUCA.

RONALD BRAVO RESTREPO



CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
POPAYÁN
2021

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN QUE PERMITA APOYAR
EL PROCESO DE SELECCIÓN DE BENEFICIARIOS DE PROYECTOS PÚBLICOS,
EJECUTADOS POR LA ALCALDÍA DEL MUNICIPIO DE TIMBÍO – CAUCA.



RONALD BRAVO RESTREPO

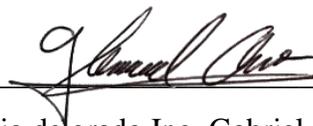
Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero de Sistemas

DIRECTOR
INGENIERO GABRIEL OSORIO

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
POPAYÁN
2021

NOTA DE ACEPTACIÓN

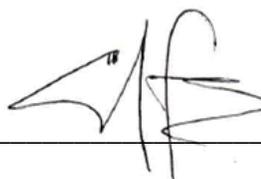
El director y jurados del trabajo de grado “IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN QUE PERMITA APOYAR EL PROCESO DE SELECCIÓN DE BENEFICIARIOS DE PROYECTOS PÚBLICOS, EJECUTADOS POR LA ALCALDÍA DEL MUNICIPIO DE TIMBÍO – CAUCA, una vez revisado el documento final y aprobada la sustentación del mismo, autorizan las gestiones administrativas correspondientes al título profesional.



Director trabajo de grado Ing. Gabriel Osorio



Jurado



Jurado

Dedicatoria

Con todo mi corazón dedico este logro a mi hermana Alexandra Bravo Restrepo por su sacrificio y esfuerzo, a mi hijo Jerónimo por ser el motor de mi vida, a mi madre Lida María Restrepo y a mi hermana Nathaly Bravo Restrepo por su amor y apoyo incondicional, y a todas las personas que de una u otra forma hicieron parte de este proceso.

Agradecimientos

Agradezco a la corporación universitaria Autónoma del Cauca y a sus docentes por su profesionalismo, por compartir sus conocimientos y brindar una educación de calidad.

Agradezco a mi director de tesis, el ingeniero GABRIEL OSORIO, por su orientación en el desarrollo del proyecto.

Agradezco al jurado por su dedicación en la revisión del informe final y a todas las personas que aportaron su grano de arena en este proceso.

TABLA DE CONTENIDO

RESUMEN	1
INTRODUCCIÓN	3
1. PROBLEMA	4
1.1. PLANTEAMIENTO DEL PROBLEMA	4
1.2. JUSTIFICACIÓN	5
1.3. OBJETIVOS	6
1.3.1. OBJETIVO GENERAL	6
1.3.2. OBJETIVOS ESPECÍFICOS	6
2. MARCO TEÓRICO	6
2.1. ANTECEDENTES	6
2.1.1. SISBÉN APP:	7
2.1.2. PROSPERIDAD SOCIAL (Familias en acción):	7
2.1.3. REGISTRO ÚNICO DE VICTIMAS (RUV):	8
2.1.4. VIVANTO:	8
2.1.5. Sistema de Afiliación Trasnacional (SAT):	8
2.1.6. SUIFP - TERRITORIO:	9
2.2. BASES TEÓRICAS	9
2.2.1. Sistemas de información:	9
2.2.2. La web	10
2.2.3. Bases de datos	10
2.2.4. Modelo Vista Controlador (MVC)	11
2.2.5. Portabilidad	11
2.2.6. Usabilidad	11
2.2.7. Metodologías ágiles	12
2.2.8. Metodología ágil Programación Extrema (XP)	12

3. METODOLOGÍA	13
3.1. METODOLOGÍA XP	13
3.1.1. Fases de la Programación Extrema	13
3.1.1.1. Planeación:	13
3.1.1.2. Diseño	15
El diseño de la aplicación evolucionará durante el transcurso de las iteraciones.	
Durante cada iteración se complementarán los siguientes diagramas.	15
3.1.1.3. Codificación	15
3.1.1.4. Pruebas	16
4. INGENIERÍA	16
4.1. HISTORIAS DE USUARIO	16
4.1.1. Como usuario administrador, necesita una lista con los proyectos creados para consultar la información almacenada en la aplicación.	16
4.1.1.1. Descripción	16
4.1.1.2. Criterios de Aceptación	17
4.1.2. Como usuario administrador, necesita crear un proyecto para gestionar beneficios.	18
4.1.2.1. Descripción	18
4.1.2.2. Criterios de Aceptación	19
4.1.3. Como usuario administrador, necesita actualizar un proyecto para gestionar beneficios	22
4.1.3.1. Descripción	23
4.1.3.2. Criterios de Aceptación	24
4.1.6. Como usuario administrador, necesita actualizar una convocatoria para asignarla a un proyecto	27
4.1.6.1. Descripción	27
4.1.6.2. Criterios de aceptación	28
4.1.4 Como usuario administrador, necesita una lista con las convocatorias creadas para consultar la información almacenada	30
4.1.4.1. Descripción	30

4.1.4.2. Criterios de Aceptación:	31
4.1.5. Como usuario administrador, necesita crear una convocatoria para asignarla a un proyecto.	32
4.1.5.1. Descripción	32
4.1.5.2. Criterios de Aceptación	33
4.1.7. Como usuario administrador, necesita adicionar un requisito para definir las condiciones de una convocatoria	34
4.1.7.1. Descripción	35
4.1.7.2. Criterios de Aceptación	36
4.1.8. Como usuario administrador, necesita una lista con los requisitos de una convocatoria para consultar la información.	40
4.1.8.1. Descripción	40
4.1.8.2. Criterios de Aceptación	41
4.1.9. Como usuario administrador, necesita eliminar un requisito para que no esté disponible en una convocatoria	42
4.1.9.1. Descripción	42
4.1.9.2. Criterios de Aceptación:	42
4.1.10. Como usuario administrador, necesita registrar un ciudadano para inscribirlo en una convocatoria	43
4.1.10.1. Descripción	43
4.1.10.2. Criterios de Aceptación	45
4.1.11. Como ciudadano, necesita registrarse en la aplicación para inscribirse en una convocatoria	50
4.1.11.1. Descripción	50
4.1.11.2. Criterios de Aceptación	52
4.1.12. Como usuario administrador necesita ver la lista de ciudadanos inscritos en una convocatoria para consultar la información	57
4.1.12.1. Descripción	57
4.1.12.2. Criterios de Aceptación:	59
4.1.13. Como ciudadano, necesita ver los resultados de la convocatoria para verificar si fue aceptado	60

4.1.13.1. Descripción	60
4.1.13.2. Criterios de Aceptación:	60
4.1.14. Como usuario administrador, necesita buscar un ciudadano en la aplicación para consultar sus datos o actualizar la información	61
4.1.14.1. Descripción	61
4.1.14.2. Criterios de Aceptación:	62
4.1.15. Como usuario administrador necesito modificar la información de un ciudadano para mantener los datos actualizados.	63
4.1.15.1. Descripción	63
4.1.15.2. Criterios de Aceptación:	66
4.1.16. Como usuario administrador, necesita ver los beneficiarios de un proyecto para consultar la información.	71
4.1.16.1. Descripción	71
4.1.16.2. Criterios de Aceptación	72
4.1.17. Como usuario administrador, necesita revisar los requisitos de un ciudadano inscrito en una convocatoria para validar la información	73
4.1.17.1. Descripción	73
4.1.17.2. Criterios de Aceptación:	74
4.1.18. Como usuario administrador, necesita inscribir un ciudadano en una convocatoria para que pueda participar en la selección de beneficiarios para un proyecto	76
4.1.18.1. Descripción	76
4.1.18.2. Criterios de Aceptación:	78
4.1.19. Como ciudadano necesita registrarse en una convocatoria para poder participar en la selección de beneficiarios para un proyecto.	81
4.1.19.1. Descripción	81
4.1.19.2. Criterios de Aceptación:	82
4.1.20. Como ciudadano, necesita cargar los documentos para poder cumplir con los requisitos exigidos por la convocatoria	84
4.1.20.1. Descripción	84
4.1.20.2. Criterios de Aceptación:	85

4.1.21. Como usuario administrador, necesita generar los beneficiarios de una convocatoria para asignarlos a un proyecto.	86
4.1.21.1. Descripción	86
4.1.21.2. Criterios de Aceptación:	87
4.1.23. Portabilidad	88
4.1.24. Usabilidad	89
4.1.25. Auditoria	89
4.2. ARQUITECTURA DE LA APLICACIÓN	89
4.3. COMPONENTES DE DESARROLLO	91
4.4. BASE DE DATOS	93
4.4.1. Diccionario de datos	94
5. HERRAMIENTAS UTILIZADAS	99
5.1. Visual Studio 2019	99
5.2. Microsoft SQL Server	100
5.3. Postman	100
5.4. Azure devops	100
5.5. React js	100
6. RESULTADOS	100
6.1 EVIDENCIAS	101
6.1.1. Formulario registrar proyectos	101
6.1.2. Formulario Registrar Convocatoria	101
6.1.3. Formulario Registrar Ciudadano	102
6.1.4. Pantalla de inicio plataforma sisebpp	103
6.1.7. Usuario Logueado Plataforma sisebppWeb	105
7. CONCLUSIONES Y RECOMENDACIONES	107
7.1. CONCLUSIONES	107

7.2. RECOMENDACIONES	108
8. BIBLIOGRAFÍA	109

LISTA DE TABLAS

Tabla 1 Tipo de Requisito.....	35
Tabla 2 Arquitectura de la aplicación	91
Tabla 3 Componentes de Desarrollo	92

LISTA DE ILUSTRACIONES

Ilustración 1 Prototipo listado de proyectos	17
Ilustración 2 Prototipo Crear proyecto	19
Ilustración 3 Prototipo Actualizar Proyecto	24
Ilustración 4 Prototipo Actualizar Convocatoria	28
Ilustración 5 Prototipo listado de convocatorias	31
Ilustración 6 Prototipo crear convocatoria	33
Ilustración 7 Prototipo adicionar requisito	36
Ilustración 8 Prototipo listado requisitos	41
Ilustración 9 Prototipo Registrar Ciudadano	45
Ilustración 10 Prototipo formulario Ciudadanos	52
Ilustración 11 Prototipo listado ciudadanos inscritos	58
Ilustración 12 Prototipo buscar ciudadano	62
Ilustración 13 Prototipo actualizar información del ciudadano	65
Ilustración 14	65
Ilustración 15 Prototipo listado beneficiarios	72
Ilustración 16 Prototipo Inscribir ciudadano en una convocatoria	78
Ilustración 17 Prototipo inscribirse en convocatoria	82
Ilustración 18 Prototipo cargar requisitos	85
Ilustración 19 Diagrama Arquitectura de la Aplicación	90
Ilustración 20 diagrama Componentes de Desarrollo	92
Ilustración 21 Diagrama Entidad Relación	93
Ilustración 22 Tabla BarriosVeredas	94
Ilustración 23 Tabla Ciudadanos	94
Ilustración 24 Tabla Convocatoria	95
Ilustración 25 Tabla ConvocatoriaCiudadano	95
Ilustración 26 Tabla Departamentos	96
Ilustración 27 Tabla Dependencias	96
Ilustración 28 Tabla Entidades	96
Ilustración 29 EstadoCiudadano	97
Ilustración 30 Tabla Municipios	97

Ilustración 31 Tabla Proyectos	97
Ilustración 32 Tabla Requisitos	98
Ilustración 33 Tabla Sector	98
Ilustración 34 Tabla TipoRequisito	99
Ilustración 35 Tabla RequisitosCiudadano	99
Ilustración 36 Formulario Registrar Proyectos	101
Ilustración 37 Formulario Registrar Proyectos	101
Ilustración 38 Formulario Registrar Convocatoria	101
Ilustración 39 Formulario Registrar Ciudadano	102
Ilustración 40 Formulario Registrar Ciudadano	102
Ilustración 41 Pantalla de inicio plataforma sisebpp	103
Ilustración 42 Pantalla de inicio plataforma sisebpp	103
Ilustración 43 Formulario Iniciar Sesión	104
Ilustración 44 Formulario de Registro	104
Ilustración 45 Formulario de Registro	105
Ilustración 46 formulario de Registro	105
Ilustración 47 Usuario Logueado Plataforma sisebppWeb	106
Ilustración 48 Usuario Logueado Plataforma sisebppWeb	106

RESUMEN

El siguiente documento pretende mostrar el diseño de una aplicación web, que servirá como apoyo a las dependencias del municipio de Timbío-Cauca en el proceso de selección de los beneficiarios de proyectos; esta nueva aplicación permitirá a las diferentes dependencias, la gestión de las convocatorias de proyectos, permitiéndoles, administrar los principales datos de la convocatoria, y los requisitos técnicos, sociales y jurídicos exigidos para participar, así como realizar consultas y generar informes sobre proyectos finalizados, proyectos en curso, convocatorias vigentes y convocatorias cerradas.

Además, les facilitará a los ciudadanos interesados consultar en el sitio web las convocatorias vigentes, inscribirse, cargar los documentos requeridos y verificar si fueron o no aceptados en los proyectos, y la razón en caso de ser negada la solicitud. Finalmente, la consolidación de la información de los proyectos y sus beneficiarios, las convocatorias y sus inscritos, permitirá cruzar la información y generar un informe de las personas que hayan sido beneficiadas en más de un proyecto y de potenciales beneficiarios para procesos de focalización

Palabras Claves: aplicación Web, beneficiarios de proyectos, convocatorias, consolidación, focalización.

ABSTRACT

The following document aims to show the design of a web application, which will serve as a support to the offices of the municipality of Timbío-Cauca in the process of selecting project beneficiaries; this new application will allow the different offices to manage the calls for projects, allowing them to manage the main data of the call, and the technical, social and legal requirements required to participate, as well as to make consultations and generate reports on completed projects, ongoing projects, current calls and closed calls.

In addition, it will make it easier for interested citizens to consult the current calls on the website, register, upload the required documents and check whether or not they were accepted into the projects, and the reason if the application is rejected. Finally, the consolidation of information on the projects and their beneficiaries, the calls for proposals and their registrants, will allow information to be cross-referred and a report to be produced on the persons who have benefited from more than one project and on potential beneficiaries for targeting processes.

Keyword: Web application, project beneficiaries, calls, consolidation, focus.

INTRODUCCIÓN

Debido a los altos niveles de desigualdad que existen en Colombia, los entes territoriales a nivel nacional, departamental y municipal tienen la obligación de distribuir los recursos públicos transformados en proyectos productivos, educativos, de obra pública, de mejoramiento de vivienda etc, de una manera equitativa entre la población más vulnerable.

El presente trabajo de investigación recurre a las aplicaciones web, teniendo en cuenta que en la actualidad son de gran importancia en cualquier entidad sea pública o privada dado que permiten establecer un canal de comunicación directo con sus clientes.

El trabajo de investigación tuvo como objetivo, desarrollar una aplicación web, para apoyar a las dependencias del municipio de Timbío-Cauca en el proceso de selección de los beneficiarios de proyectos, esto con el fin de poder distribuir los recursos entre la población de una forma más equitativa, así como facilitarle a la comunidad, sobre todo a las personas que se encuentran más alejadas del sector urbano, por medio de la plataforma, acceder a las convocatorias y poder inscribirse desde cualquier lugar, y evitar así el desplazamiento hasta las instalaciones de la alcaldía del municipio de Timbío, para realizar dicho trámite.

El trabajo se estructura de la siguiente manera, en primer lugar, se presenta el campo problemático, donde se hace hincapié en las dificultades que presentan los funcionarios de la alcaldía del municipio de Timbío en la selección de los de los beneficiarios de los proyectos. Segundo, se presenta el marco de referencia en el cual se abordan los principales referentes teóricos conceptuales y la metodología utilizada. Tercero, se muestran los resultados y desemboca en las conclusiones y recomendaciones.

1. PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La Alcaldía del municipio de Timbío - Cauca, viene realizando un gran esfuerzo para que los recursos se distribuyan de la mejor manera entre la población más vulnerable; la entidad enfrenta una dificultad con la selección de los beneficiarios, que ocasiona que algunos ciudadanos reciban beneficios de diferentes proyectos, quitándole a otros la posibilidad de participar.

Actualmente cada dependencia gestiona los proyectos de manera autónoma; cuando se crea una convocatoria para un nuevo proyecto, esta es publicada en la radio local y a través de avisos en las entidades y oficinas del municipio, para que la población interesada se dirija a las instalaciones de la alcaldía a realizar el proceso de inscripción. Dicha inscripción se realiza en la dependencia encargada de ejecutar el proyecto y los ciudadanos interesados entregan físicamente los documentos exigidos para participar en la convocatoria, así la información de los ciudadanos inscritos es almacenada en un archivo Excel por cada convocatoria, de tal manera que las dependencias del municipio cuentan con decenas de archivos de beneficiarios de proyectos ejecutados y de ciudadanos inscritos a nuevas convocatorias.

Los funcionarios responsables de cada convocatoria realizan manualmente el proceso de selección de beneficiarios, normalmente deben dirigirse a solicitar la información a otras dependencias para realizar el cruce de los datos y no incurrir principalmente en el error de seleccionar personas que hayan sido objeto de ayudas de otros programas o proyectos y de esta manera limitar la posibilidad de favorecer a otras.

Además de cruzar la información con archivos de otras dependencias, también deben cruzar la información con archivos de su propia dependencia, lo anterior ha generado lentitud de respuesta en los procesos e inconsistencias en la selección de los beneficiarios, complica el manejo de la información por no encontrarse centralizada y no existen protocolos que garanticen la seguridad de los datos.

Es así como surge la idea de desarrollar una aplicación web, para apoyar a las dependencias del municipio de Timbío-Cauca en el proceso de selección de los beneficiarios de proyectos; esta nueva aplicación permitirá a las diferentes dependencias, la gestión de las convocatorias de proyectos, permitiéndoles, administrar los principales datos de la convocatoria, y los requisitos técnicos, sociales y jurídicos exigidos para participar, así como realizar consultas y generar informes sobre proyectos finalizados, proyectos en curso, convocatorias vigentes y convocatorias cerradas.

Adicional, les facilitará a los ciudadanos interesados consultar en el sitio web las convocatorias vigentes, inscribirse, cargar los documentos requeridos y verificar si fueron o no aceptados en los proyectos, y la razón en caso de ser negada la solicitud. Finalmente, la consolidación de la información de los proyectos y sus beneficiarios, las convocatorias y sus inscritos, permitirá cruzar la información y generar un informe de las personas que hayan sido beneficiadas en más de un proyecto y de potenciales beneficiarios para procesos de focalización, de tal manera que permita a cada una de las dependencias tomar la mejor decisión y garantizar que las personas seleccionadas son aptas para acceder a los beneficios.

1.2. JUSTIFICACIÓN

El trabajo que las entidades gubernamentales realicen en sus regiones y la forma en como estos distribuyen los recursos entre la población más vulnerable es de vital importancia para el desarrollo de los territorios y el progreso de su población.

La implementación de un sistema de información para la Alcaldía del municipio de Timbío-Cauca se realizará con el fin de contribuir en la mejora del proceso de selección de las personas beneficiarias de proyectos financiados con recursos públicos, en cuanto a la consolidación de la información de los proyectos y sus beneficiarios y de las personas inscritas a las nuevas convocatorias, de esta manera la aplicación permitirá realizar el cruce de la información y generar un informe de las personas que hayan sido beneficiadas en más de un proyecto y de potenciales beneficiarios para procesos de focalización, evitando así desplazarse por cada una de las dependencias extrayendo la información histórica para realizar el cruce de la misma, que en muchos casos no se encuentra o está incompleta y de incurrir en el error de seleccionar personas que hayan sido objeto de ayudas de otros proyectos.

La aplicación permitirá a las dependencias administrar la información de las convocatorias y de los requisitos que se exigen para participar en cada una de estas, como realizar consultas y generar informes sobre proyectos finalizados, proyectos en curso, convocatorias vigentes y convocatorias cerradas, así como permitir a la comunidad ingresar, registrarse, consultar e inscribirse a las convocatorias a través del aplicativo y verificar si fueron o no aceptados y las razones en caso de ser negada la solicitud.

Siendo la seguridad de la información pilar de toda empresa, el traslado de esta a un Sistema Gestor de Base de Datos robusto y potente tipo SQL Server que permitirá establecer niveles de seguridad, con el control de ingreso de la información (y hacia ella) y se podrá interactuar con el sistema desde cualquier lugar, pues se podrá alojar la base de datos en un servidor para garantizar la concurrencia de usuarios.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Implementar un sistema de información que permita apoyar el proceso de selección de los beneficiarios de proyectos públicos, ejecutados por la alcaldía del municipio de Timbío – cauca.

1.3.2. OBJETIVOS ESPECÍFICOS

- Determinar los requerimientos necesarios para el desarrollo del sistema de selección de beneficiarios.
- Desarrollar la arquitectura del sistema, basada en la especificación de requisitos.
- Evaluar el sistema de información con un caso de estudio en la alcaldía del municipio de Timbio-Cauca.

2. MARCO TEÓRICO

2.1. ANTECEDENTES

El desarrollo de aplicaciones para el manejo de programas sociales ha sido un tema de gran importancia para el gobierno nacional en Colombia. Por ello existen diversas aplicaciones que permiten clasificar a la población más vulnerable del país teniendo en cuenta algunos aspectos

sociales y económicos, y de esta manera poder otorgarle ciertos beneficios a esta parte de la población.

2.1.1. SISBÉN APP:

Sisbén APP es una aplicación que permite clasificar a las personas asignándoles un puntaje, de acuerdo con su estrato y condición social, este puntaje es uno de los principales requisitos para poder acceder a cualquier programa, proyecto o beneficio que entreguen los entes territoriales a la población más vulnerable.

El Sisbén es un registro de potenciales beneficiarios de programas sociales al que puede ingresar cualquier colombiano. Su objetivo principal es ordenar a la población mediante un puntaje de acuerdo con sus características, y así identificar los potenciales beneficiarios de la oferta social. Al ser una herramienta de focalización individual, es un insumo importante para la oferta social en la entrega de subsidios y beneficios a la población (Declaración de importancia Estratégica del Sistema de Identificación de Potenciales Beneficiarios (SISBEN IV), 2016, pág. 10).

2.1.2. PROSPERIDAD SOCIAL (Familias en acción):

El programa de familias en acción cuenta con una plataforma web que permite clasificar a los niños, niñas y adolescentes menores de 18 años en condiciones de pobreza y que son beneficiados con subsidios orientados al crecimiento, desarrollo y asistencia escolar.

Más Familias en Acción - MFA es un programa de Transferencias Monetarias Condicionadas TMC que inicia su operación a finales del año 2000, con el fin de mitigar el impacto de la recesión económica sobre las familias más pobre del país. Orientado a la entrega de incentivos condicionados a las asistencias a las atenciones en salud individuales de los niños y niñas en primera infancia y a la asistencia escolar de los niños, niñas y adolescentes en edad escolar (Colombia. Departamento Administrativo para la Prosperidad Social, 2019, pág. 4).

Actualmente el programa es una política de Estado, enmarcado en la Ley 1532 de 2012, siendo uno de los principales programas de promoción social del país, en un contexto de corresponsabilidades compartidas entre las familias participantes del programa y el estado*4 (Colombia. Departamento Administrativo para la Prosperidad Social, 2019, pág. 4).

2.1.3. REGISTRO ÚNICO DE VÍCTIMAS (RUV):

Es una herramienta que permite registrar y clasificar a las personas víctimas de la violencia en Colombia, y de esta manera poder reparar a esta parte parte de la población.

Herramienta administrativa que soporta el procedimiento de registro de las víctimas, la cual permite la identificación de la población que ha sufrido algún daño o vulneración de sus derechos de acuerdo con lo establecido en el artículo 3° de la Ley 1448 de 2011 (Colombia. Unidad para las Víctimas , 2018).

2.1.4. VIVANTO:

Es un sistema de información web y en línea que permite la consulta consolidada de la información relativa a una víctima, desde sus datos básicos, declaraciones presentadas bajo cualquier marco normativo que conforman el Registro Único de Víctimas, hasta los beneficios recibidos de los diferentes programas del Gobierno Nacional. A través de la Red Nacional de Información, se consolida toda la información de los diferentes sistemas de las entidades del SNARIV y de los cuatro marcos normativos que conforman el RUV (SIPOD, SIV, SIRAV y ley 1448 de 2011), con las restricciones de seguridad y confidencialidad de la información respectiva (Colombia. Unidad para las Víctimas , 2018).

2.1.5. Sistema de Afiliación Transnacional (SAT):

Esta plataforma permite el registro y actualización de la información de los colombianos al Sistema General de Seguridad Social en Salud y permite clasificar a las personas en condiciones de pobreza de acuerdo al puntaje del SISBEN para garantizarles el derecho a la salud.

La herramienta permite realizar los procesos de afiliación y novedades y definir los instrumentos para garantizar a la comunidad la afiliación y el goce efectivo del derecho a la salud.

El aseguramiento se realiza de manera general a través de las EPS del régimen contributivo y subsidiado y, por tanto, los ciudadanos tienen el deber de asegurar su permanencia en uno u otro régimen según corresponda. Así, el Sistema General de Seguridad Social en salud garantiza el acceso a los servicios indistintamente de la capacidad de pago, y aquellas personas que no la

tienen pueden afiliarse al régimen subsidiado (Colombia. Ministerio de Salud y Protección Social, 2015, pág. 2).

2.1.6. SUIFP - TERRITORIO:

El Sistema Unificado de Inversiones y Finanzas Públicas es un sistema de información que integra los procesos asociados a cada una de las fases del ciclo de la inversión pública, acompañando los proyectos de inversión desde su formulación hasta la entrega de los productos, articulándolos con los programas de gobierno y las políticas públicas.

A través del SUIFP, se realizará el ingreso de la información mínima requerida en los proyectos de inversión y se soportarán los procesos para gestionar el proyecto en sus etapas de formulación, presentación, viabilidad, ejecución, incluyendo su seguimiento y evaluación. Sin embargo, la gestión de los proyectos se continuará desarrollando de acuerdo con los procedimientos y normatividad establecidos para cada fuente de financiación (Colombia. Departamento Nacional de Planeación, 2018, pág. 6).

Lo antes expuesto, indica que el aprovechamiento de las tecnologías de la información (TIC) en la administración pública actual es de vital importancia ya que mejora la calidad de los servicios públicos, profundiza la transparencia de su actuación y promueve la participación ciudadana en la formulación, ejecución, control y evaluación de los planes y programas de desarrollo orientados a mejorar las condiciones de vida de la población a nivel nacional, regional y local.

2.2. BASES TEÓRICAS

2.2.1. Sistemas de información:

Los Sistemas de Información Gerencial (SIG) pueden definirse como un conjunto integrado de componentes, que tiene el objetivo de recolectar, almacenar, procesar y proporcionar datos y cualquier otro tipo de producto digital. Los componentes principales de un sistema de información son: (1) el hardware o componentes físicos, (2) el software o código fuente, (3) las telecomunicaciones, (4) bases de datos y servidores, y (5) recursos humanos y procedimientos. En la estructura organizacional de una empresa, los SIG se localizan en un nivel administrativo, en el que se relaciona con funciones de seguimiento, control, toma de decisiones y

administración de recursos. Se encuentra en dicho nivel ya que la información que produce ejerce la función de herramienta de planificación y toma de decisiones que sirve de ayuda para la mejora de la productividad de una organización, usando así sus operaciones diarias de control. Los sistemas que se encuentran en un nivel administrativo tienen como objetivo proporcionar informes o resúmenes regulares relacionados con las operaciones o procedimientos que ejecuta la empresa. (Proaño Castro, Orellana Contreras, & Martillo Pazmillo, 2018).

2.2.2. La web

En la web 1.0 las páginas web son estáticas y con poca participación de los usuarios en la elaboración de los contenidos, lo que hace que el uso de la web sea más pasivo que activo. Estas páginas contenían pocas imágenes y todavía menos elementos multimedia como video o audio, debido a la gran cantidad de espacio de almacenamiento que requerían y a los enlaces de baja velocidad existentes para conectarse a la internet (Becerril Isidro, Vallejo Lassard, Lumbreras Sotomayor, Chávez Ojeda, & Duk Sánchez, 2012, pág. 24).

En la web 2.0 se establece el uso para mantener relaciones sociales de manera virtual, logrando con ello desaparecer las distancias y las limitantes del tiempo en estas relaciones, ejemplos de ello se encuentran en múltiples aplicaciones on-line tales como Facebook, twitter, hi5, etcétera. Es por esto que a la versión 2.0 se le llama la web social o colaborativa (Becerril Isidro, Vallejo Lassard, Lumbreras Sotomayor, Chávez Ojeda, & Duk Sánchez, 2012, pág. 25).

2.2.3. Bases de datos

En la actualidad, las bases de datos se usan tan ampliamente que se pueden encontrar en organizaciones de todos los tamaños, desde grandes corporaciones y agencias gubernamentales, hasta pequeños negocios e incluso en hogares. Las actividades diarias con frecuencia lo ponen en contacto con las bases de datos, ya sea directa o indirectamente (Ricardo, 2009, pág. 2).

las bases de datos se usan para satisfacer las necesidades de información de muchas organizaciones e individuos en una variedad de áreas. Sin embargo, una base de datos deficientemente diseñada fracasa para proporcionar la información requerida u ofrece información no actualizada, falsa o contradictoria. Con la finalidad de maximizar sus beneficios

potenciales, es importante comprender los fundamentos teóricos, estructura interna, diseño y gestión de las bases de datos (Ricardo, 2009, pág. 3).

2.2.4. Modelo Vista Controlador (MVC)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman **Modelos, Vistas y Controladores** (Desarrolloweb, 2020).

2.2.5. Portabilidad

Se define como la característica que posee un software para ejecutarse en diferentes plataformas, es decir, la portabilidad se refiere exclusivamente a la propiedad que posee un software que le permite ser ejecutado en diferentes plataformas y/o sistemas operativos. De este modo, si un determinado software compilado pudiese ser ejecutado en cualquier sistema operativo, diríamos que ese software es 100% portable (Wikipedia, 2021).

2.2.6. Usabilidad

es un atributo cualitativo definido comúnmente como la facilidad de uso, ya sea de una página Web, una aplicación informática o cualquier otro sistema que interactúe con un usuario. El concepto generalmente se refiere a una aplicación informática o un aparato, aunque también puede aplicarse a cualquier sistema hecho con algún objetivo particular. También se refiere a métodos para mejorar la facilidad de uso durante el proceso de diseño (Sanchez, 2011, pág. 8).

2.2.7. Metodologías ágiles

El termino metodologías ágiles se hizo popular cuando diecisiete especialistas en desarrollo de software, presentaron los métodos Extreme Programming (XP), Scrum, Feature Driven, Development FDD, entre otros, establecieron principios comunes compartidos por todos. El resultado fu la creación de la Alianza (Agile Allince) y el establecimiento del manifiesto Ágil o Agile Manifiesto, en el año de 2005. Las metodologías ágiles varían en sus prácticas y en sus fases, sin embargo, comparten algunas características tales como: desarrollo iterativo e incremental, comunicación y reducción de productos intermediarios y de la documentación extensiva (LAÍNES Fuentes, 2015, pág. 104).

Durante 1996, Beck es llamado por Chrysler como asesor del proyecto Chrysler Comprehensive Compensation (C3) payroll system. Dada la poca calidad del sistema que se estaba desarrollando, Beck decide tirar todo el código y empezar de cero utilizando las prácticas que él había ido definiendo a lo largo del tiempo. El sistema, que administra la liquidación de aproximadamente 10.000 empleados, y consiste de 2.000 clases y 30.000 métodos, es puesto en operación en mayo de 1997 casi respetando el calendario propuesto. Como consecuencia del éxito de dicho proyecto, Kent Beck dio origen a XP iniciando el movimiento de metodologías ágiles al que se anexarían otras metodologías surgidas mucho antes que el propio Beck fuera convocado por Chrysler (Amaro Calderón & Valverde Rebaza, 2007, pág. 7).

2.2.8. Metodología ágil Programación Extrema (XP)

La metodología XP se considera una metodología leve de desarrollo de software. Esta es clasificada como un sistema de prácticas que la comunidad de desarrolladores de software viene evolucionando para resolver los problemas de entrega de software de calidad rápidamente, y poder alcanzar las necesidades de negocio que siempre cambian. Esta surgió partir de ideas de Kent Beck y Ward Cunningham y que fue utilizada por primera vez en un proyecto piloto en marzo de 1996, del cual el propio Beck formaba parte (LAÍNES Fuentes, 2015, pág. 107).

La XP no se aplica a todos los tipos de proyectos, siendo más apropiada para los proyectos con equipos pequeños o medianos, de dos a doce personas. Sin embargo, algunos defienden su uso en grandes proyectos, ya que al dividirlos en subproyectos independientes. Los proyectos largos

deben ser partidos en una secuencia de mini proyectos de auto contenidos de una o dos semanas (LAÍNES Fuentes, 2015, pág. 107).

Según Teles la XP es un proceso de desarrollo de software apropiado para los siguientes proyectos:

- Requisitos que son vagos y que cambian con frecuencia.
- Desarrollo de sistemas orientados a objetos.
- Equipos pequeños.
- Desarrollo incremental. Para el autor XP está organizada para asegurar que el cliente siempre reciba un alto retorno de la inversión del software.

3. METODOLOGÍA

3.1. METODOLOGÍA XP

3.1.1. Fases de la Programación Extrema

La Programación Extrema consta de 4 fases, las cuales son:

3.1.1.1. Planeación:

Teniendo en cuenta que es un trabajo de grado tipo pasantía, el equipo estará conformado por:

- El estudiante, quien tendrá a cargo los roles de gestor, líder funcional, programador y tester,
- El cliente, quien será el responsable de definir y priorizar las historias de usuario.

El proyecto se desarrollará en iteraciones de tres semanas cada una y se trabajará una iteración cero, para listar la totalidad de historias de usuario de los siguientes procesos:

- Gestión de usuarios
- Gestión de proyectos
- Gestión de convocatorias
- Gestión de requisitos
- Gestión de requisitos por convocatoria
- Inscripción de los ciudadanos y cargue de documentos exigidos para cada convocatoria.
- Proceso de selección de beneficiarios para un proyecto
- Reportes

- Cargue de la información de los proyectos y sus beneficiarios, la cual se encuentra registrada en los archivos Excel de cada dependencia de la alcaldía del municipio de Timbío - Cauca. Este cargue de información se realizará de forma manual por medio de la base de datos, debido a que es un proceso que solo se ejecutará una vez para ingresar la información al sistema. Posteriormente los usuarios continuarán el registro de información a través de la aplicación.

Sin embargo, durante la ejecución de las iteraciones se refinará la lista de historias de usuario y la priorización. La iteración cero también tendrá como objetivo priorizar las primeras historias de usuario a implementar, realizar la definición de estas y escribir el detalle con sus respectivos criterios de aceptación.

Posteriormente durante cada iteración se realizarán las siguientes actividades:

Detalle Historias de Usuarios: para esto se realizarán reuniones con el cliente con el objetivo de definir el alcance funcional de las historias de usuario priorizadas.

Posterior a las reuniones con el cliente, el Líder Funcional (Estudiante) realizará el detalle de las historias de usuario con sus respectivos criterios de aceptación.

Las historias de usuario serán escritas en lenguaje natural, con descripciones cortas de lo que el sistema debe realizar. También se utilizarán bosquejos o prototipos que le permitan al usuario identificar las funcionalidades a implementar.

Plan de Entregas (Release Plan): se realizará una reunión entre el estudiante y el cliente, para seleccionar las historias de usuario que hayan sido priorizadas y que formaran parte de la iteración.

El estudiante definirá las tareas técnicas que se requieren para implementar y probar cada historia de usuario, con el fin de determinar la complejidad.

Cuando el estudiante ya tenga definida la complejidad de las historias de usuario se comprometerá con el cliente sobre cuales historias de usuario se implementarán en la iteración.

La selección de historias de usuario para la iteración dependerá del tiempo estimado para las actividades de desarrollo y pruebas.

Esta reunión se llevará a cabo el primer día de cada iteración.

Plan de Iteraciones (Iteration Plan): el estudiante iniciará el desarrollo y las pruebas de las historias de usuario comprometidas para la iteración.

Reuniones Diarias de Seguimiento (Stand – Up Meeting), se realizarán reuniones diarias entre el estudiante y el cliente con el objetivo de compartir problemas a tiempo y proponer soluciones.

Al final de cada iteración (3 semanas) se desplegará el incremento del producto en el ambiente de pruebas, para que el usuario pueda validar las historias de usuario implementadas.

Cuando varios incrementos del producto ya tengan valor funcional para el cliente, se realizará el despliegue en el ambiente de producción.

3.1.1.2. Diseño

El diseño de la aplicación evolucionará durante el transcurso de las iteraciones. Durante cada iteración se complementarán los siguientes diagramas.

- Diagrama de base de datos.
- Diagrama de arquitectura.
- Diagrama de infraestructura.
- Diagramas de componentes.

Adicional, cuando se presenten problemas técnicos que dificulten estimar el tiempo para implementar una historia de usuario, se usarán pequeños programas de prueba llamados “**Spike**”, para explorar diferentes soluciones técnicas.

3.1.1.3. Codificación

Para la implementación de la solución se usarán las siguientes actividades, las cuales son recomendadas por la metodología XP:

Uso de Estándares, XP promueve la programación basada en estándares, de manera que sea fácil de entender para todo el equipo, y que facilite la recodificación. Durante el proyecto se utilizarán estándares de nombramiento y comentarios en el código, y estándares de nombramiento en las tablas y columnas de la base de datos.

Programación Dirigida por las Pruebas (“Test-Driven Programming”): durante la implementación se utilizará este modelo, donde primero se escriben las pruebas unitarias y posteriormente se desarrollan las funcionalidades las cuales deben pasar las pruebas previamente definidas. Esta práctica garantiza la calidad del código, debido a que crear diferentes casos de pruebas unitarias aumenta la cobertura del código.

Control de Código Fuente: se utilizará Git para versionar los cambios efectuados en el código.

3.1.1.4. Pruebas

Pruebas Unitarias: al final de cada iteración se garantizará que las funcionalidades implementadas pasen las pruebas unitarias para liberar la versión al cliente.

Detección y Corrección de Errores: si el cliente encuentra errores críticos durante las pruebas, estos serán solucionados durante la iteración, de lo contrario serán priorizados en la reunión (**Release Plan**), para decidir si se incluyen en la siguiente iteración.

Pruebas de Aceptación: de acuerdo con los criterios de aceptación definidos en cada historia de usuario, se crearán los casos de pruebas manuales, para ser ejecutadas en cada iteración, con el fin de comprobar que las historias de usuario comprometidas en la iteración han sido implementadas correctamente.

4. INGENIERÍA

4.1. HISTORIAS DE USUARIO

Las **historias de usuario**, son pequeñas descripciones de los requerimientos de un cliente. Su utilización es común cuando se aplica marcos de entornos ágiles como XP. Al redactar las **historias de usuario** se debe tener en cuenta describir el Rol, la funcionalidad y el resultado esperado en una frase corta

4.1.1. Como usuario administrador, necesita una lista con los proyectos creados para consultar la información almacenada en la aplicación.

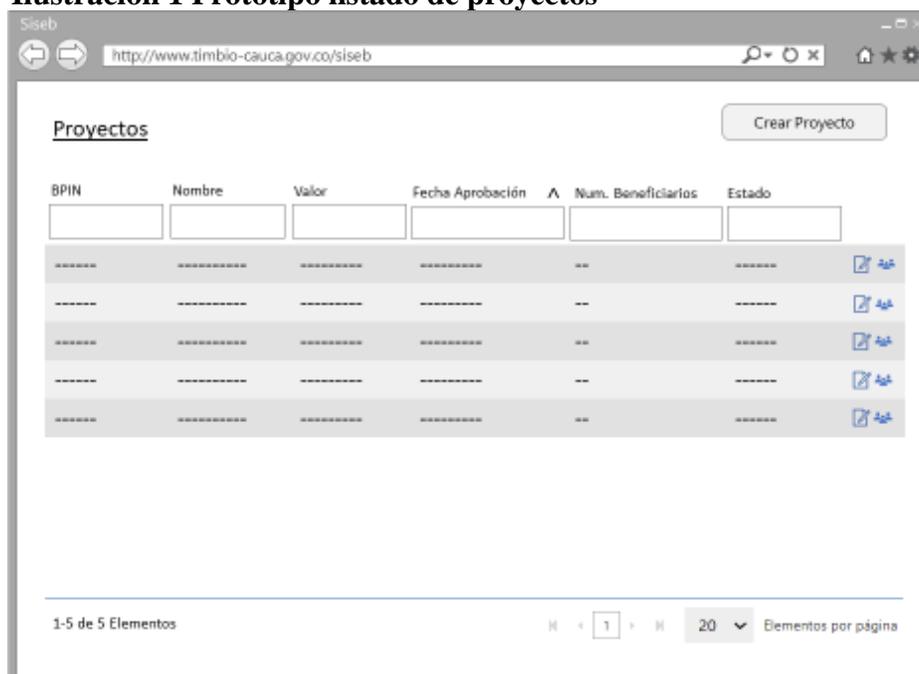
4.1.1.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar en una grilla los proyectos almacenados. La grilla deberá mostrar los siguientes datos:

- BPIN: código BPIN del proyecto.
- Nombre: nombre del proyecto.
- Valor: valor del proyecto.
- Fecha Aprobación.
- Número Beneficiarios.
- Estado: puede contener los valores "activo, inactivo"

Los proyectos deben mostrarse ordenados por fecha de creación del más reciente al más antiguo.

Ilustración 1 Prototipo listado de proyectos



Fuente: Investigación propia

4.1.1.2. Criterios de Aceptación

AC_01: Existen proyectos

Dado que: el usuario necesita ver los proyectos creados en la aplicación.

Cuando: existen proyectos creados y el usuario hace clic en la opción de menú Proyectos.

Entonces:

- La aplicación deberá mostrar en una grilla los proyectos creados.
- La información que se debe mostrar por cada proyecto debe ser "BPIN", "Nombre", "Valor", "Fecha Aprobación", "Número Beneficiarios", "Estado".
- Los proyectos deben mostrarse ordenados por la fecha de creación del más reciente al más antiguo.

AC_02: No existen proyectos creados

Dado que: el usuario necesita ver los proyectos creados en la aplicación.

Cuando: NO existen proyectos creados y el usuario hace clic en la opción de menú Proyectos.

Entonces: la aplicación deberá mostrar el texto "Sin registros" debajo de los nombres de las columnas de la grilla.

4.1.2. Como usuario administrador, necesita crear un proyecto para gestionar beneficios.

4.1.2.1. Descripción

Actualmente la oficina de planeación de la Alcaldía es la encargada de los proyectos del municipio que ofrecen beneficios a los ciudadanos. El objetivo de esta historia de usuario consiste en crear un proyecto para posteriormente abrir una convocatoria y seleccionar sus beneficiarios.

El alcance de la historia de usuario no contempla la gestión administrativa de un proyecto.

En la lista "proyectos" debe existir un botón para crear un nuevo proyecto. Cuando el usuario de clic en este botón, la aplicación deberá abrir el formulario para crear un nuevo proyecto.

Un proyecto debe tener la siguiente información:

- Código BPIN: No puede existir más de un proyecto con el mismo número. El código del proyecto debe contener letras y números sin espacios y caracteres especiales, y puede contener hasta 15 caracteres.
- Nombre del proyecto: El nombre de un proyecto puede contener todo tipo de caracteres y el número máximo debe ser 200 caracteres.

- Valor del Proyecto: el valor del proyecto debe ser un número decimal con máximo dos decimales.
- Número de documento de aprobación: no puede existir más de un proyecto con el mismo número. El código del proyecto debe contener letras y números sin espacios y caracteres especiales, y puede contener hasta 15 caracteres.
- Fecha de aprobación: la fecha de aprobación debe ser menor o igual a la fecha actual
- Sector: debe permitir seleccionar el sector al que pertenece el proyecto (agrícola, agua potable, infraestructura).
- Dependencia: se debe permitir seleccionar la dependencia que va a ejecutar el proyecto (planeación, agricultura, cultura)
- Número beneficiarios: se debe poder ingresar cualquier número entero positivo.
- Observaciones: este campo puede contener todo tipo de caracteres y el número máximo debe ser 200. Este dato es opcional.
- Estado: por defecto un proyecto debe crearse en estado activo.

Nota: todos los datos del proyecto excepto las observaciones son de carácter obligatorio.

Ilustración 2 Prototipo Crear proyecto

The screenshot shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. The main content is a form titled "Crear Proyecto" with a blue header and a close button (X). The form contains the following fields:

- Código BPIN:** A text input field.
- Nombre:** A text input field.
- Valor:** A text input field.
- Número de Documento:** A text input field.
- Fecha de Aprobación:** A text input field.
- Sector:** A dropdown menu with the text "Seleccionar >>" and a downward arrow.
- Dependencia:** A dropdown menu with the text "Seleccionar >>" and a downward arrow.
- Número de beneficiarios:** A text input field.
- Observaciones:** A large text area with the placeholder text "Digite el objetivo y/o descripción del proyecto".

At the bottom right of the form, there are two buttons: "Cancelar" and "Guardar".

Fuente: Investigación propia

4.1.2.2. Criterios de Aceptación

AC_01: Abrir formulario proyectos

Dado que: el usuario administrador se encuentra en la lista "proyectos".

Cuando: el usuario de clic en el botón "Nuevo Proyecto".

Entonces: la aplicación debe abrir el formulario de proyectos para poder crearlo.

AC_02: El código BPIN ya existe

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario ingresa un código BPIN que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el código BPIN ya existe "El código BPIN ingresado ya se encuentra registrado en la aplicación.".

AC_03: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_04: Fecha de aprobación no puede ser mayor a la fecha actual

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario da clic en el campo fecha.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_05: Formato Código BPIN

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite el código.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y solo debe permitir 15 caracteres.

AC_06: Formato Nombre del Proyecto

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite el Nombre del Proyecto.

Entonces: el campo debe permitir ingresar cualquier tipo de carácter y máximo 200.

AC_07: Formato Valor del Proyecto

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite el valor del proyecto

Entonces: el campo debe permitir ingresar solo números positivos con máximo dos decimales.

AC_08: Formato número de aprobación

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite el número de aprobación.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y debe permitir solo 10 caracteres.

AC_09: Número de aprobación ya existe

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario ingresa un número de aprobación que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el número de aprobación ya existe "El número de aprobación ingresado ya se encuentra registrado en la aplicación."

AC_10: Listado de sectores

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario haga clic sobre la lista de sectores.

Entonces: la aplicación deberá mostrar un listado de los sectores almacenados en la base de datos.

AC_11: Listado de dependencias

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario haga clic sobre la lista de dependencias.

Entonces: la aplicación deberá mostrar un listado de las dependencias almacenadas en la base de datos.

AC_12: Campo número de beneficiarios

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite el número de beneficiarios.

Entonces: el campo debe permitir ingresar solo números enteros positivos.

AC_13: Campo observaciones

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario digite una observación.

Entonces: el campo debe permitir cualquier tipo de carácter y máximo 200.

AC_14: Campo estado

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario haga clic sobre el botón guardar

Entonces: la aplicación debe registrar el proyecto en estado activo.

AC_15: Almacenar un proyecto

Dado que: el usuario administrador se encuentra en el formulario para crear un proyecto.

Cuando: el usuario haga clic sobre el botón guardar

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de proyectos el proyecto creado. El nuevo proyecto deberá aparecer de primero en la lista.

4.1.3. Como usuario administrador, necesita actualizar un proyecto para gestionar beneficios

4.1.3.1. Descripción

Actualmente la oficina de planeación de la Alcaldía es la encargada de los proyectos del municipio que ofrecen beneficios a los ciudadanos. El objetivo de esta historia de usuario consiste en actualizar un proyecto.

En la lista de proyectos, cada registro de un proyecto debe tener una opción editar, que permita abrir el formulario de actualización de la información.

Todos los campos del formulario deben mostrarse con la información del proyecto a actualizar.

Un proyecto debe tener la siguiente información:

- **Código BPIN:** este campo debe ser de solo lectura
- **Nombre del proyecto:** El nombre de un proyecto puede contener todo tipo de caracteres y el número máximo debe ser 200 caracteres.
- **Valor del Proyecto:** el valor del proyecto debe ser un número decimal con máximo dos decimales.
- **Número de documento de aprobación:** no puede existir más de un proyecto con el mismo número. El código del proyecto debe contener letras y números sin espacios y caracteres especiales, y puede contener hasta 15 caracteres.
- **Fecha de aprobación:** debe permitir cambiar la fecha de aprobación. La fecha debe ser menor o igual a la fecha actual.
- **Sector:** debe permitir seleccionar el sector al que pertenece el proyecto (agrícola, agua potable, infraestructura).
- **Dependencia:** se debe permitir seleccionar la dependencia que va a ejecutar el proyecto (planeación, agricultura, cultura).
- **Número beneficiarios:** se debe poder ingresar cualquier número entero positivo.
- **Observaciones:** este campo puede contener todo tipo de caracteres y el número máximo debe ser 200. Este dato es opcional.
- **Estado:** debe permitir seleccionar el estado el cual puede ser activo o inactivo.

Nota: todos los datos del proyecto excepto las observaciones son de carácter obligatorio.

Ilustración 3 Prototipo Actualizar Proyecto

The screenshot shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. The main content is a modal window titled 'Actualizar Proyecto'. The form inside has the following fields:

- Código BPIN:** A text input field that is disabled (greyed out).
- Nombre:** A text input field.
- Valor:** A text input field.
- Número de Documento:** A text input field.
- Fecha de Aprobación:** A text input field.
- Sector:** A dropdown menu.
- Dependencia:** A dropdown menu.
- Número de beneficiarios:** A text input field.
- Estado:** A dropdown menu.
- Observaciones:** A large text area for notes.

At the bottom right of the form are two buttons: 'Cancelar' and 'Guardar'.

Fuente: investigación propia

4.1.3.2. Criterios de Aceptación

AC_01: mostrar información del proyecto a actualizar

Dado que: el usuario administrador se encuentra en la página con la lista de proyectos.

Cuando: el usuario hace clic en la opción editar del proyecto que quiere actualizar.

Entonces:

- La aplicación debe abrir el formulario de actualización de proyectos, mostrar la información del proyecto y permitir su actualización.
- Todos los campos son actualizables a excepción del código BPIN que debe ser de solo lectura.

AC_02: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_03: Fecha de aprobación no puede ser mayor a la fecha actual

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario da clic en el campo fecha.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_04: Formato Nombre del Proyecto

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario digite el Nombre del Proyecto.

Entonces: el campo debe permitir ingresar cualquier tipo de carácter y máximo 200.

AC_05: Formato Valor del Proyecto

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario digite el valor del proyecto.

Entonces: el campo debe permitir ingresar solo números positivos con máximo dos decimales.

AC_06: Formato número de aprobación

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario digite el número de aprobación.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y debe permitir solo 10 caracteres.

AC_07: Número de aprobación ya existe

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario actualiza el número de aprobación del proyecto por otro que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el número de aprobación ya existe "El número de aprobación ingresado ya se encuentra registrado en la aplicación".

AC_08: Listado de sectores

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario haga clic sobre la lista de sectores.

Entonces: la aplicación deberá mostrar un listado de los sectores almacenados en la base de datos y por defecto debe mostrar seleccionado el sector del proyecto.

AC_09: Listado de dependencias

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario haga clic sobre la lista de dependencias.

Entonces: la aplicación deberá mostrar un listado de las dependencias almacenadas en la base de datos y por defecto debe mostrar seleccionada la dependencia ejecutora del proyecto.

AC_10: Campo número de beneficiarios

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario digite el número de beneficiarios.

Entonces: el campo debe permitir ingresar solo números enteros positivos.

AC_11: Campo observaciones

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario digite una observación.

Entonces: el campo debe permitir cualquier tipo de carácter y máximo 200.

AC_12: Campo estado

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario haga clic sobre la lista estado.

Entonces: la aplicación debe mostrar los estados "activo, inactivo" y por defecto debe mostrar el estado del proyecto.

AC_13: Actualizar un proyecto

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de un proyecto.

Cuando: el usuario haga clic sobre el botón guardar.

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de proyectos el proyecto con la información actualizada.

4.1.6. Como usuario administrador, necesita actualizar una convocatoria para asignarla a un proyecto

4.1.6.1. Descripción

El objetivo de esta historia de usuario consiste en actualizar la información de la convocatoria.

En la lista de convocatorias, cada registro de una convocatoria debe tener una opción editar, que permita abrir el formulario de actualización de la información.

Todos los campos del formulario deben mostrarse con la información de la convocatoria a actualizar.

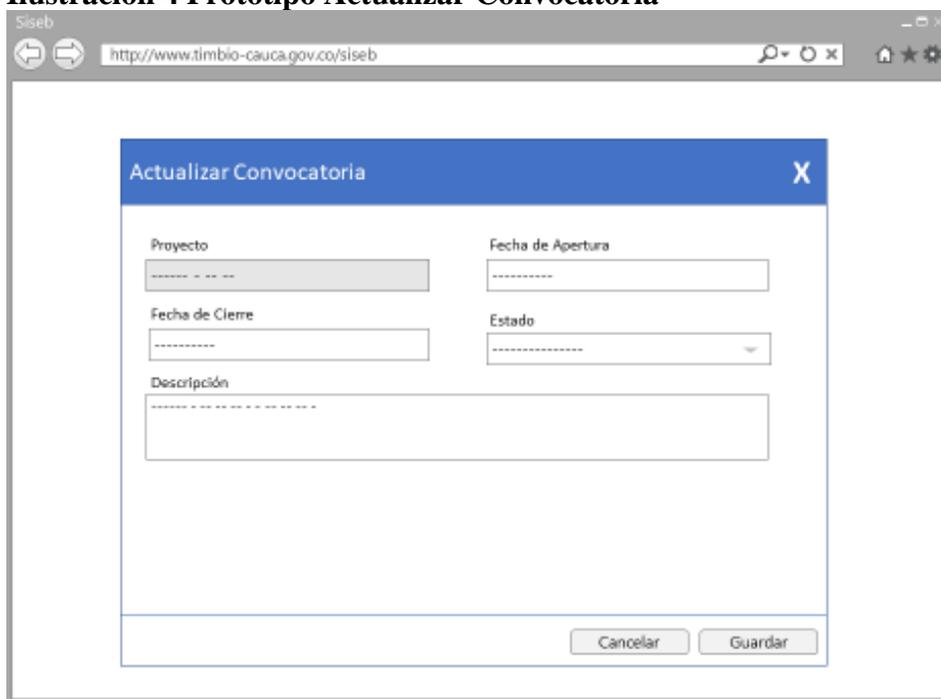
La convocatoria debe tener la siguiente información:

- Proyecto: este campo debe ser de solo lectura y mostrar el proyecto al que pertenece la convocatoria.

- Descripción: debe permitir cualquier tipo de carácter y puede contener hasta 200 caracteres.
- Fecha de Apertura: debe permitir cambiar la fecha de apertura. La fecha debe ser menor o igual a la fecha de cierre de la convocatoria.
- Fecha de cierre: debe permitir cambiar la fecha de cierre. La fecha debe ser mayor o igual a la fecha de apertura de la convocatoria.
- Estado: debe permitir seccionar el estado, el cual puede ser abierta, cerrada o inactiva.

Nota: todos los datos de la convocatoria son de carácter obligatorio.

Ilustración 4 Prototipo Actualizar Convocatoria



Actualizar Convocatoria

Proyecto:

Fecha de Apertura:

Fecha de Cierre:

Estado:

Descripción:

Cancelar Guardar

Fuente: investigación propia

4.1.6.2. Criterios de aceptación

AC_01: Mostrar información de la convocatoria a actualizar

Dado que: el usuario administrador se encuentra en la página con la lista de las convocatorias.

Cuando: el usuario hace clic en la opción editar de la convocatoria que quiere actualizar.

Entonces:

- La aplicación debe abrir el formulario de actualización de la convocatoria, mostrar la información de la convocatoria y permitir su actualización.
- El campo proyecto debe ser de solo lectura y mostrar el proyecto al cual pertenece la convocatoria.

AC_02: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para actualizar una convocatoria.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_03: Campo descripción

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de una convocatoria.

Cuando: el usuario digite la información de la descripción de la convocatoria.

Entonces: la aplicación debe permitir ingresar cualquier tipo de carácter y máximo 200 caracteres.

AC_04: Listado de proyectos

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de una convocatoria.

Cuando: el usuario haga clic sobre la lista de los proyectos.

Entonces: la aplicación deberá mostrar un listado de los proyectos vigentes almacenados en la base de datos.

AC_05: Fecha de apertura debe ser menor o igual a la fecha de cierre

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de una convocatoria.

Cuando: el usuario seleccione una fecha de apertura mayor a la fecha de cierre y haga clic sobre el botón guardar.

Entonces: la aplicación debe mostrar un mensaje "la fecha de apertura debe ser menor o igual a la fecha de cierre".

AC_06: Campo estado

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de una convocatoria.

Cuando: el usuario haga clic sobre la lista estado.

Entonces: la aplicación debe mostrar los estados "abierta, cerrada, inactiva" y por defecto debe mostrar el estado de la convocatoria.

AC_07: Actualizar convocatoria

Dado que: el usuario administrador se encuentra en el formulario para actualizar la información de una convocatoria.

Cuando: el usuario haga clic sobre el botón guardar.

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de convocatorias, la convocatoria con la información actualizada.

4.1.4 Como usuario administrador, necesita una lista con las convocatorias creadas para consultar la información almacenada

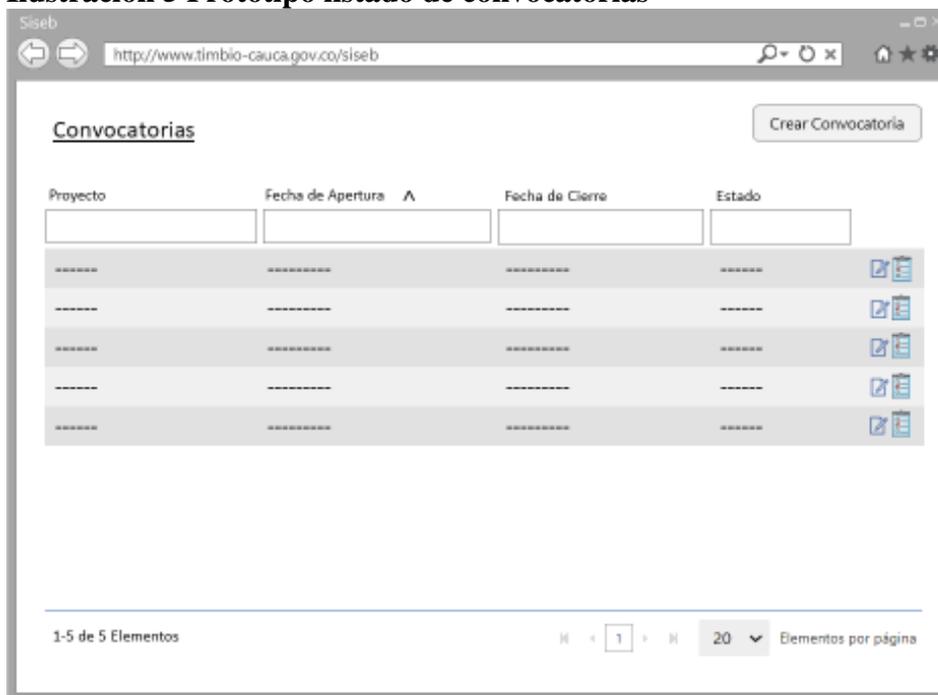
4.1.4.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar en una grilla las convocatorias almacenadas. La grilla deberá mostrar los siguientes datos:

- Proyecto: nombre del proyecto.
- Fecha de Apertura: fecha de aprobación.
- Fecha de Cierre: fecha de cierre.
- Estado: puede contener los valores "abierto, cerrado, inactivo"

Las convocatorias deben mostrarse ordenadas por fecha de creación de la más reciente a la más antigua.

Ilustración 5 Prototipo listado de convocatorias



Fuente: investigación propia

4.1.4.2. Criterios de Aceptación:

AC_01: Existen convocatorias

Dado que: el usuario administrador necesita ver las convocatorias creadas en la aplicación.

Cuando: existen convocatorias creadas y el usuario hace clic en la opción de menú convocatoria.

Entonces:

- La aplicación deberá mostrar en una grilla de las convocatorias creadas.
- La información que se debe mostrar por cada convocatoria debe ser "Proyecto", "Fecha de apertura", "Fecha de Cierre", "Estado".
- Las convocatorias deben mostrarse ordenados por la fecha de creación del más reciente al más antiguo.

AC_02: No existen convocatorias creadas

Dado que: el usuario administrador necesita ver las convocatorias creadas en la aplicación.

Cuando: NO existen convocatorias creadas y el usuario hace clic en la opción de menú Convocatoria.

Entonces: La aplicación deberá mostrar el texto "Sin registros" debajo de los nombres de las columnas de la grilla.

4.1.5. Como usuario administrador, necesita crear una convocatoria para asignarla a un proyecto.

4.1.5.1. Descripción

El objetivo de esta historia de usuario consiste en crear una convocatoria para un proyecto.

En la lista "Convocatorias" debe existir un botón para crear una nueva convocatoria. Cuando el usuario de clic en este botón, la aplicación deberá abrir el formulario para crear una nueva convocatoria.

La lista convocatoria debe tener la siguiente información:

- Proyecto: debe permitir seleccionar cualquiera de los proyectos vigentes creados en la aplicación.
- Fecha de Apertura: la fecha de apertura debe ser menor o igual a la fecha de cierre de la convocatoria
- Fecha de cierre: la fecha de cierre debe ser mayor o igual a la fecha de apertura de la convocatoria.
- Estado: por defecto una convocatoria debe crearse en estado abierto.
- Descripción: debe permitir ingresar cualquier tipo de carácter y puede contener hasta 200 caracteres.

Nota: todos los datos de la convocatoria son de carácter obligatorio.

Ilustración 6 Prototipo crear convocatoria

The image shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. The browser title is 'Siseb'. The main content is a modal window titled 'Crear Convocatoria' with a close button (X). The form contains the following fields:

- Proyecto:** A dropdown menu with the text 'Seleccionar >>'.
- Fecha de Apertura:** A text input field.
- Fecha de Cierre:** A text input field.
- Descripción:** A larger text input field with the placeholder text 'Digite el objetivo y/o descripción de la convocatoria'.

At the bottom right of the form are two buttons: 'Cancelar' and 'Guardar'.

Fuente: investigación propia

4.1.5.2. Criterios de Aceptación

AC_01: Abrir formulario convocatoria

Dado que: el usuario administrador se encuentra en la lista "convocatorias".

Cuando: el usuario de clic en el botón "Nueva Convocatoria".

Entonces: la aplicación debe abrir el formulario de convocatorias para poder crearla.

AC_02: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_03: Campo descripción

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el usuario digite la información de la descripción de la convocatoria.

Entonces: la aplicación debe permitir ingresar cualquier tipo de carácter y máximo 200 caracteres.

AC_04: Listado de proyectos

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el usuario haga clic sobre la lista de los proyectos.

Entonces: la aplicación deberá mostrar un listado de los proyectos vigentes almacenados en la base de datos.

AC_05: Fecha de apertura debe ser menor o igual a la fecha de cierre

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el usuario seleccione una fecha de apertura mayor a la fecha de cierre y haga clic sobre el botón guardar.

Entonces: la aplicación debe mostrar un mensaje "la fecha de apertura debe ser menor o igual a la fecha de cierre".

AC_06: Almacenar convocatoria

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el usuario haga clic sobre el botón guardar.

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de convocatorias la convocatoria creada. La nueva convocatoria deberá aparecer de primera en la lista.

AC_07: Campo estado

Dado que: el usuario administrador se encuentra en el formulario para crear una convocatoria.

Cuando: el usuario haga clic sobre el botón guardar.

Entonces: la aplicación debe registrar la convocatoria en estado abierta.

4.1.7. Como usuario administrador, necesita adicionar un requisito para definir las condiciones de una convocatoria

4.1.7.1. Descripción

El objetivo de esta historia de usuario consiste en agregar un requisito a una convocatoria.

La página que contiene la lista de requisitos, deberá tener un botón para adicionar un nuevo requisito.

El formulario de adicionar requisito debe tener la siguiente información:

- Nombre: puede contener todo tipo de caracteres y máximo se deben permitir 50.
- Tipo de Requisito: debe mostrar una lista con las opciones "Documento", "Sisbén", "Estado Civil", "Cabeza Familia", "Fecha Nacimiento", "Zona", "Estrato". Cada tipo de requisito tiene definido un tipo de dato:

Tabla 1 Tipo de Requisito

Tipo de requisito	Tipo de dato	Lista de información
Documento	Documento	
Sisbén	Decimal	
Estado Civil	Lista	Soltero(a), Casado(a), Divorciado(a), Unión Libre, Viudo(a)
Cabeza Familia	Booleano	
Fecha Nacimiento	Fecha	
Zona	Lista	Rural, Urbano
Estrato	Entero	

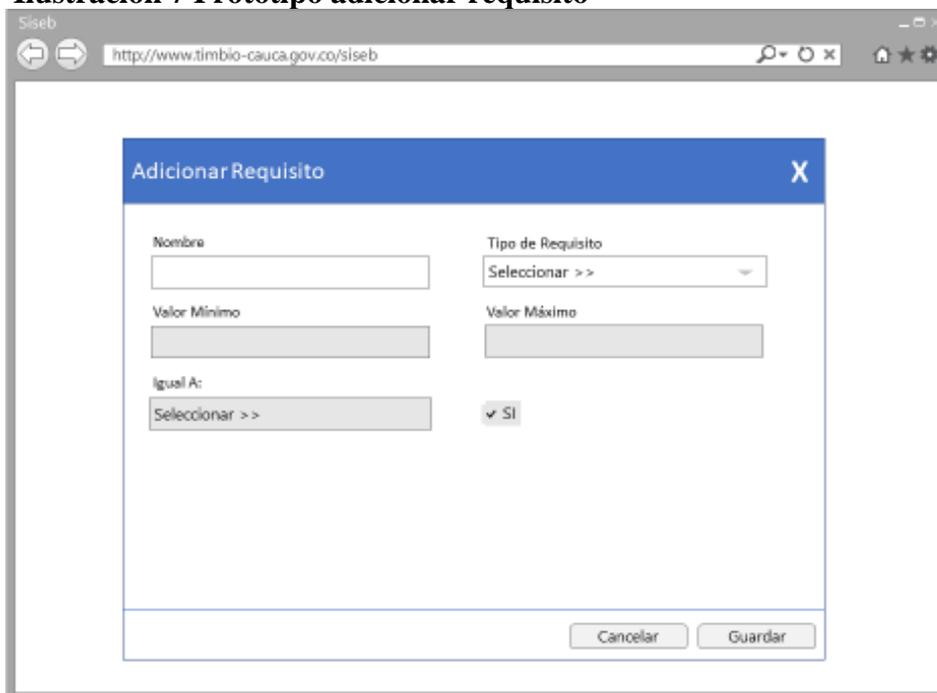
Fuente: investigación propia

- Cuando el usuario seleccione de la lista una opción, donde el tipo de dato sea igual a **Decimal**, la aplicación deberá habilitar los campos "Valor Mínimo" y "Valor Máximo", los cuales solo deben permitir números positivos con máximo dos decimales. El valor máximo debe ser mayor o igual al valor mínimo y los dos valores son de carácter obligatorio.
- Cuando el usuario seleccione de la lista una opción, donde el tipo de dato sea igual a **Entero**, la aplicación deberá habilitar los campos "Valor Mínimo" y "Valor Máximo", los cuales solo

deben permitir números enteros positivos. El valor máximo debe ser mayor o igual al valor mínimo y los dos valores son de carácter obligatorio.

- Cuando el usuario seleccione de la lista una opción, donde el tipo de dato sea igual a **Fecha**, la aplicación deberá habilitar los campos "Valor Mínimo" y "Valor Máximo", los cuales deberán tener un calendario para seleccionar las fechas. La fecha seleccionada en el valor máximo debe ser mayor o igual a la fecha seleccionada en el valor mínimo y los dos valores son de carácter obligatorio.
- Cuando el usuario seleccione de la lista una opción, donde el tipo de dato sea igual a **Booleano**, la aplicación deberá habilitar un control para que el usuario seleccione SI o NO.
- Cuando el usuario seleccione de la lista una opción, donde el tipo de dato sea igual a **Lista**, la aplicación deberá habilitar una lista de selección múltiple con las opciones configuradas. El usuario debe seleccionar por los menos una opción de la lista.

Ilustración 7 Prototipo adicionar requisito



The image shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. A modal dialog box titled "Adicionar Requisito" is open. The dialog contains the following fields and controls:

- Nombre:** A text input field.
- Tipo de Requisito:** A dropdown menu with the text "Seleccionar >>".
- Valor Mínimo:** A text input field.
- Valor Máximo:** A text input field.
- Igual A:** A dropdown menu with the text "Seleccionar >>".
- SI:** A radio button that is currently selected.

At the bottom of the dialog, there are two buttons: "Cancelar" and "Guardar".

Fuente: investigación propia

4.1.7.2. Criterios de Aceptación

AC_01: Abrir el formulario de requisitos

Dado que: el usuario administrador se encuentra en la página con la lista de requisitos.

Cuando: el usuario hace clic sobre el botón "Adicionar Requisito".

Entonces:

- La aplicación debe abrir el formulario para agregar un nuevo requisito.
- La lista tipo de requisito debe mostrar por defecto la opción "Seleccionar >>".
- Los controles "Valor Mínimo", "Valor Máximo", "Igual A", y "SI" deberán estar deshabilitados

AC_02: Formato nombre del requisito

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario digite el Nombre del requisito.

Entonces: el campo debe permitir ingresar cualquier tipo de carácter y máximo 50.

AC_03: Listado de tipos de requisitos

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario haga clic sobre la lista Tipo.

Entonces: la aplicación deberá mostrar un listado con las opciones "Documento", "Sisbén", "Estado Civil", "Cabeza Familia", "Fecha Nacimiento", "Zona", "Estrato".

AC_04: Tipo de dato igual a documento

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Documento".

Entonces: la aplicación debe deshabilitar los controles "Valor Mínimo", "Valor Máximo", "Igual A", y "SI".

AC_05: Tipo de dato igual a decimal

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Decimal".

Entonces:

- La aplicación debe habilitar los controles "Valor Mínimo" y "Valor Máximo".
- Los controles "Valor Mínimo" y "Valor Máximo" solo deben permitir el ingreso de números positivos con máximo dos decimales.
- La aplicación debe deshabilitar los controles "Igual A" y "SI".

AC_06: Tipo de dato igual a entero

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Entero".

Entonces:

- La aplicación debe habilitar los controles "Valor Mínimo" y "Valor Máximo".
- Los controles "Valor Mínimo" y "Valor Máximo" solo deben permitir el ingreso de números enteros positivos.
- La aplicación debe deshabilitar los controles "Igual A" y "SI".

AC_07: Tipo de dato igual a fecha

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Fecha".

Entonces:

- La aplicación debe habilitar los controles "Valor Mínimo" y "Valor Máximo".
- Los controles "Valor Mínimo" y "Valor Máximo" deben tener un calendario para seleccionar fechas.
- La aplicación debe deshabilitar los controles "Igual A" y "SI".

AC_08: Tipo de dato igual a booleano

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Booleano".

Entonces:

- La aplicación debe habilitar el control "SI", el cual debe ser de tipo check box.
- La aplicación debe deshabilitar los controles "Valor Mínimo", "Valor Máximo" e "Igual A".

AC_09: Tipo de dato igual a lista

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario selecciona una opción de la lista "Tipo de requisito" y el tipo de dato del elemento seleccionado es igual a "Lista".

Entonces:

- La aplicación debe habilitar el control "Igual A".
- La lista "Igual A" debe contener las opciones configuradas para el tipo de requisito y debe permitir seleccionar más de un elemento.
- La aplicación debe deshabilitar los controles "Valor Mínimo", "Valor Máximo" y "SI".

AC_10: El valor mínimo debe ser mayor o igual al valor máximo.

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el valor mínimo es mayor que el valor máximo y el usuario hace clic sobre el botón "Guardar".

Entonces: la aplicación deberá mostrar debajo del campo "Valor Mínimo" el mensaje "El valor mínimo debe ser menor o igual que el valor máximo".

AC_11: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_12: Almacenar un requisito

Dado que: el usuario administrador se encuentra en el formulario para agregar un requisito.

Cuando: el usuario haga clic sobre el botón guardar

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de requisitos, el requisito creado. El nuevo requisito deberá aparecer de primero en la lista.

4.1.8. Como usuario administrador, necesita una lista con los requisitos de una convocatoria para consultar la información.

4.1.8.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar en una grilla, los requisitos creados para una convocatoria.

Desde la lista de convocatorias, el usuario deberá tener una opción para ver los requisitos de una convocatoria.

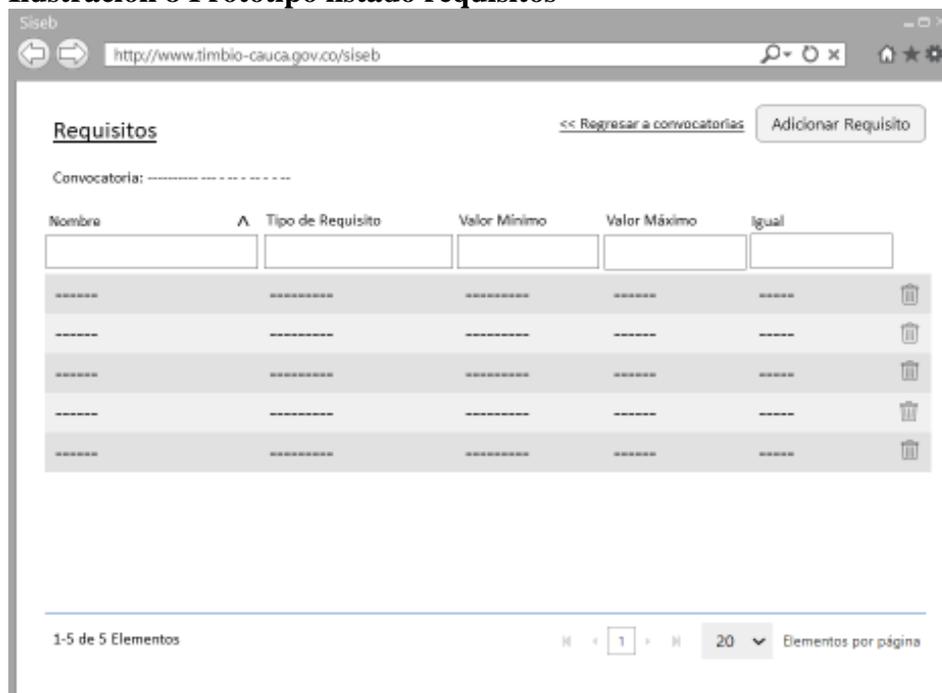
Cuando el usuario haga clic en la opción requisitos de una convocatoria, la aplicación deberá abrir una página para ver los requisitos.

La página debe mostrar el nombre de la convocatoria, un link para volver a la lista de convocatorias y una grilla con los requisitos de la convocatoria. La grilla de requisitos debe mostrar la siguiente información:

- Nombre: nombre del requisito.
- Tipo de Requisito: nombre del tipo de requisito, el cual puede ser "Documento", "Sisbén", "Estado Civil", "Cabeza Familia", "Fecha Nacimiento", "Zona".
- Valor Mínimo: valor establecido como mínimo para el requisito, si aplica.
- Valor Máximo: valor establecido como máximo para el requisito, si aplica.
- Igual: valor establecido como igual para el requisito, si aplica.

Los requisitos deben mostrarse ordenados por fecha de creación, del más reciente al más antiguo.

Ilustración 8 Prototipo listado requisitos



Fuente: investigación propia

4.1.8.2. Criterios de Aceptación

AC_01: La convocatoria no tiene requisitos

Dado que: el usuario administrador se encuentra en la página con la lista de convocatorias.

Cuando: el usuario hace clic en la opción "Requisitos" de una convocatoria y la convocatoria NO tiene ningún requisito.

Entonces: La aplicación deberá abrir la página de requisitos, mostrar el nombre de la convocatoria y en la grilla deberá aparecer el texto "Sin registros" debajo de los nombres de las columnas.

AC_02: Ver los requisitos de una convocatoria

Dado que: el usuario administrador se encuentra en la página con la lista de convocatorias.

Cuando: el usuario hace clic en la opción "Requisitos" de una convocatoria y la convocatoria ya tiene requisitos definidos.

Entonces:

- La aplicación deberá abrir la página de requisitos de una convocatoria.

- La aplicación deberá mostrar el nombre de la convocatoria.
- La aplicación deberá mostrar la grilla con la información de los requisitos. los datos que se deben mostrar por cada requisito son "Nombre", "Tipo de Requisito", "Valor Mínimo", "Valor Máximo", "Igual".
- Los requisitos deben visualizarse ordenados por fecha de creación, del más reciente al más antiguo.

AC_03: Volver a la lista de convocatorias

Dado que: el usuario administrador se encuentra en la página de requisitos de una convocatoria.

Cuando: el usuario hace clic sobre el link "Regresar a convocatorias".

Entonces: la aplicación deberá retornar a la página con la lista de convocatorias.

4.1.9. Como usuario administrador, necesita eliminar un requisito para que no esté disponible en una convocatoria

4.1.9.1. Descripción

El objetivo de esta historia de usuario consiste en eliminar un requisito.

Cada registro del listado de requisitos de una convocatoria, debe contener una opción para eliminar la información.

4.1.9.2. Criterios de Aceptación:

AC_01: Mostrar mensaje de verificación

Dado que: el usuario administrador se encuentra en la página con la lista de requisitos de una convocatoria.

Cuando: el usuario hace clic en la opción eliminar del requisito que quiere eliminar.

Entonces:

- La aplicación debe mostrar un mensaje de verificación "está seguro que desea eliminar el requisito".

- el mensaje de verificación debe contener dos opciones "aceptar y cancelar".

AC_02: Eliminar requisito

Dado que: el usuario administrador se encuentra visualizando el mensaje de verificación para eliminar un requisito.

Cuando: el usuario hace clic en la opción "aceptar"

Entonces: la aplicación debe cerrar el mensaje de verificación y quitar el registro de la lista de requisitos.

4.1.10. Como usuario administrador, necesita registrar un ciudadano para inscribirlo en una convocatoria

4.1.10.1. Descripción

El objetivo de esta historia de usuario consiste en registrar un ciudadano para posteriormente asignarlo a una convocatoria.

En el menú debe existir una opción para registrar ciudadanos.

Un ciudadano debe tener la siguiente información:

- Tipo de identificación: debe mostrar un listado con los diferentes tipos de identificación "cedula de ciudadanía, cedula de extranjería, tarjeta de identidad, registro civil".
- Número de identificación: No puede existir más de un ciudadano con el mismo número de identificación, el número de identificación debe contener letras y números sin espacios y sin caracteres especiales, y puede contener hasta 20 caracteres.
- Primer nombre: este campo solo debe contener letras y espacios, sin caracteres especiales y puede contener hasta 50 caracteres.
- Segundo nombre: este campo solo debe contener letras y espacios, sin caracteres especiales y puede contener hasta 50 caracteres.
- Primer apellido: este campo solo debe contener letras y espacios, sin caracteres especiales y puede contener hasta 50 caracteres.

- Segundo apellido: este campo solo debe contener letras y espacios, sin caracteres especiales y puede contener hasta 50 caracteres.
- Departamento de nacimiento: debe permitir seleccionar el departamento de Colombia en el que nació el ciudadano.
- Municipio de nacimiento: debe mostrar un listado de municipios que permita seleccionar el municipio en el que nació el ciudadano y que están relacionados al departamento de nacimiento seleccionado anteriormente.
- Fecha de nacimiento: la fecha de nacimiento debe ser menor o igual a la fecha actual
- Departamento de residencia: debe permitir seleccionar el departamento de Colombia en el que vive el ciudadano.
- Municipio de residencia: debe mostrar un listado de municipios que permita seleccionar el municipio en el que vive el ciudadano, y que están relacionados al departamento de residencia seleccionado anteriormente.
- Zona de residencia: debe permitir seleccionar de una lista la zona de residencia de un ciudadano "urbana y rural"
- Estrato: número entero positivo entre 1 y 6.
- Barrio / Vereda de residencia: este campo debe permitir seleccionar la vereda o el barrio en el que vive el ciudadano y que esté relacionado con el municipio de residencia y la zona de residencia. Si anteriormente en el campo "Zona de residencia" el ciudadano selecciona "urbana", en este campo se deben mostrar los barrios del municipio y en caso contrario, si selecciona "rural" debe mostrar las veredas del municipio.
- Dirección: este campo puede contener cualquier tipo de carácter y máximo 30 caracteres.
- Teléfono Celular: este campo solo debe contener números y máximo 10 caracteres.
- Email: este campo puede contener cualquier tipo de carácter, sin espacios y debe ser formato de correo electrónico.
- Estado civil: el estado civil debe tener las opciones Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a).
- Puntaje Sisbén: este campo debe contener un número positivo con máximo dos decimales y por defecto debe contener el valor de cero.
- Cabeza familia: SI o No.

Nota: los datos deben ser obligatorios a excepción del segundo nombre, segundo apellido, Departamento de nacimiento, municipio de nacimiento, email y teléfono celular.

Ilustración 9 Prototipo Registrar Ciudadano

The screenshot shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. The page title is "Registrar Ciudadano".

Datos Personales

Tipo de Identificación Seleccionar >>	Número de Identificación [Input Field]	Primer Nombre [Input Field]
Segundo Nombre [Input Field]	Primer Apellido [Input Field]	Segundo Apellido [Input Field]
Estado Civil Seleccionar >>	Puntaje Sisbén [Input Field]	Cabeza Familia <input checked="" type="checkbox"/> SI

Datos Nacimiento

Departamento Seleccionar >>	Municipio Seleccionar >>	Fecha de Nacimiento [Input Field]
--------------------------------	-----------------------------	--------------------------------------

Datos Contacto

Departamento Seleccionar >>	Municipio Seleccionar >>	Zona Seleccionar >>
Barrio / Vereda Seleccionar >>	Estrato [Input Field]	Dirección [Input Field]
Teléfono Celular [Input Field]	Correo Electrónico [Input Field]	

Buttons: Cancelar, Guardar

Fuente: investigación propia

4.1.10.2. Criterios de Aceptación

AC_01: Abrir formulario de ciudadanos

Dado que: el usuario administrador necesita registrar un usuario en la aplicación.

Cuando: el usuario de clic en la opción de menú "Registrar Ciudadano".

Entonces: la aplicación debe abrir el formulario de Ciudadanos para poder crear uno nuevo.

AC_01: Número de identificación ya existe

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario ingresa un número de identificación que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el número de identificación ya existe "El número de identificación ya existe".

AC_02: Campos obligatorios

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_03: Fecha de nacimiento no puede ser mayor a la fecha actual

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario da clic en el campo fecha.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_04: Correo electrónico invalido

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario haya terminado de digitar el correo y si el formato es incorrecto y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el formato del correo electrónico es incorrecto "El formato del correo electrónico es incorrecto".

AC_05: listado tipo de identificación

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario haga clic sobre la lista tipo de identificación.

Entonces: la aplicación deberá mostrar un listado de los tipos de identificación disponibles en la aplicación.

AC_06: Formato número de identificación

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el número de identificación.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y solo debe permitir 20 caracteres.

AC_07: Formato primer nombre

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el primer nombre del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_08: Formato segundo nombre

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el segundo nombre del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_09: Formato primer apellido

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el primer apellido del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_10: Formato segundo apellido

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el segundo apellido del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_11: Listado departamentos de nacimiento

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_12: Listado municipios de nacimiento

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de nacimiento seleccionado anteriormente.

AC_13: Formato fecha de nacimiento

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre el campo fecha de nacimiento.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_14: Listado departamentos de residencia

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_15: Listado municipios de residencia

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de residencia seleccionado.

AC_16: Listado zonas

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista zonas.

Entonces: la aplicación deberá mostrar una lista con las opciones "urbana, rural".

AC_17: Listado barrios/veredas de residencia

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic sobre la lista barrios/veredas.

Entonces:

- La aplicación deberá mostrar un listado de los barrios o veredas disponibles en la base de datos y que estén relacionados al departamento de residencia y a la zona seleccionados.
- Si la zona de residencia seleccionada es urbana debe mostrar los barrios del municipio en caso contrario cuando es rural debe mostrar las veredas.

AC_18: Formato número de teléfono

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el número de teléfono.

Entonces: el campo debe restringir cualquier carácter excepto números y máximo 10 caracteres.

AC_19: Formato Email

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario digite el correo electrónico.

Entonces: el campo debe permitir cualquier carácter excepto espacios y máximo 50 caracteres.

AC_20: Campo estado civil

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic en la lista "estado civil"

Entonces: la aplicación debe mostrar las opciones "Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a)".

AC_21: Campo puntaje Sisbén

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el puntaje del Sisbén

Entonces: el campo debe permitir solo números positivos con máximo dos decimales

AC_22: Campo estrato

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el estrato

Entonces: el campo debe permitir solo números positivos entre 1 y 6.

AC_23: Registrar ciudadano

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario haga clic sobre el botón guardar

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de ciudadanos, el ciudadano registrado. El nuevo ciudadano deberá aparecer de primero en la lista.

4.1.11. Como ciudadano, necesita registrarse en la aplicación para inscribirse en una convocatoria

4.1.11.1. Descripción

Los ciudadanos del municipio de Timbío son los más interesados en tener la posibilidad de registrarse a través de una plataforma sin que tengan la necesidad de desplazarse hasta la alcaldía del municipio para poder inscribirse en las convocatorias. El objetivo de esta historia de usuario consiste en que un ciudadano puede registrarse en la aplicación para que posteriormente pueda inscribirse en una convocatoria.

Un ciudadano debe tener la siguiente información:

- Tipo de identificación: debe mostrar un listado con los diferentes tipos de identificación "cedula de ciudadanía, cedula de extranjería, tarjeta de identidad, registro civil".
- Número de identificación: No puede existir más de un ciudadano con el mismo número de identificación y debe ser un entero positivo. El número de identificación debe contener letras y números sin espacios y caracteres especiales, y puede contener hasta 20 caracteres.
- Primer nombre: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.

- Segundo nombre: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Primer apellido: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Segundo apellido: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Departamento de nacimiento: debe permitir seleccionar el departamento de Colombia en el que nació el ciudadano.
- Municipio de nacimiento: debe mostrar un listado de municipios que permita seleccionar el municipio en el que nació el ciudadano y que están relacionados al departamento de nacimiento seleccionado anteriormente.
- Fecha de nacimiento: la fecha de nacimiento debe ser menor o igual a la fecha actual
- Departamento de residencia: debe permitir seleccionar el departamento de Colombia en el que vive el ciudadano.
- Municipio de residencia: debe mostrar un listado de municipios que permita seleccionar el municipio en el que vive el ciudadano, y que están relacionados al departamento de residencia seleccionado.
- Zona de residencia: debe permitir seleccionar de una lista la zona de residencia de un ciudadano "urbana y rural"
- Estrato: número entero positivo entre 1 y 6.
- Barrio / Vereda de residencia: este campo debe permitir seleccionar la vereda o el barrio en el que vive el ciudadano y que esté relacionado con el municipio de residencia y la zona de residencia. Si anteriormente en el campo "Zona de residencia" el ciudadano selecciona "urbana", en este campo se deben mostrar los barrios del municipio y en caso contrario, si selecciona "rural" debe mostrar las veredas del municipio.
- Dirección: este campo puede contener cualquier tipo de carácter y máximo 30 caracteres.
- Teléfono Celular: este campo solo debe contener un número entero positivo.
- Correo Electrónico: este campo puede contener cualquier tipo de carácter, sin espacios y debe ser formato de correo electrónico.
- Estado civil: el estado civil debe tener las opciones Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a).

- Puntaje Sisbén: este campo debe contener un número positivo con máximo dos decimales y por defecto debe contener el valor de cero.
- Cabeza familia: SI o No.

Nota: los datos deben ser obligatorios a excepción del segundo nombre, segundo apellido, departamento de nacimiento, municipio de nacimiento, dirección, email y teléfono celular.

Ilustración 10 Prototipo formulario Ciudadanos

The screenshot shows a web browser window with the URL <http://www.timbio-cauca.gov.co/siseb>. The page title is "Registrar Ciudadano". The form is organized into three main sections:

- Datos Personales:** Includes fields for "Tipo de Identificación" (dropdown), "Número de Identificación" (text), "Primer Nombre" (text), "Segundo Nombre" (text), "Primer Apellido" (text), "Segundo Apellido" (text), "Estado Civil" (dropdown), "Puntaje Sisbén" (text), and "Cabeza Familia" (checkbox, checked).
- Datos Nacimiento:** Includes fields for "Departamento" (dropdown), "Municipio" (dropdown), and "Fecha de Nacimiento" (text).
- Datos Contacto:** Includes fields for "Departamento" (dropdown), "Municipio" (dropdown), "Zona" (dropdown), "Barrio / Vereda" (dropdown), "Estrato" (text), "Dirección" (text), "Teléfono Celular" (text), and "Correo Electrónico" (text).

At the bottom right of the form are "Cancelar" and "Guardar" buttons.

Fuente: investigación propia

4.1.11.2. Criterios de Aceptación

AC_01: Número de identificación ya existe

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano ingresa un número de identificación que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al ciudadano que el número de identificación ya existe "El número de identificación ya existe".

AC_02: Campos obligatorios

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_03: Fecha de nacimiento no puede ser mayor a la fecha actual

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano da clic en el campo fecha.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_04: El formato del correo electrónico no es el correcto

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano haya terminado de digitar el correo y si el formato es incorrecto y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al ciudadano que el formato del correo electrónico es incorrecto "El formato del correo electrónico es incorrecto".

AC_05: Listado tipo de identificación

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano haga clic sobre la lista tipo de identificación.

Entonces: la aplicación deberá mostrar un listado de los tipos de identificación disponibles en la aplicación.

AC_06: Formato número de identificación

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el número de identificación.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y solo debe permitir 20 caracteres.

AC_07: Formato primer nombre

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el primer nombre.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_08: Formato segundo nombre

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el segundo nombre.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_09: Formato primer apellido

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el primer apellido.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_10: Formato segundo apellido

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el segundo apellido.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_11: listado departamentos de nacimiento

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_12: Listado municipios de nacimiento

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de nacimiento seleccionado anteriormente.

AC_13: Formato fecha de nacimiento

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre el campo fecha de nacimiento.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_14: Listado departamentos de residencia

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_15: Listado municipios de residencia

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de residencia seleccionado anteriormente.

AC_16: Listado zonas

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista zonas.

Entonces: la aplicación deberá mostrar una lista con las opciones "urbana, rural".

AC_17: Listado barrios/veredas de residencia

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano de clic sobre la lista barrios/veredas.

Entonces:

- La aplicación deberá mostrar un listado de los barrios o veredas disponibles en la base de datos y que estén relacionados al departamento de residencia y a la zona seleccionados anteriormente.
- Si la zona de residencia seleccionada anteriormente es urbana debe mostrar los barrios del municipio en caso contrario si es rural debe mostrar las veredas.

AC_18: Formato número de teléfono

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el número de teléfono.

Entonces: el campo debe restringir cualquier carácter excepto números.

AC_19: Formato Email

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano digite el correo electrónico.

Entonces: el campo debe permitir cualquier carácter excepto espacios y validar que este escrito en formato de correo electrónico.

AC_20: Campo estado civil

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic en la lista "estado civil"

Entonces: la aplicación debe mostrar las opciones "Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a)".

AC_21: Campo puntaje Sisbén

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el puntaje del Sisbén

Entonces: el campo debe permitir solo números positivos con máximo dos decimales

AC_22: Campo estrato

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el estrato

Entonces: el campo debe permitir solo números positivos entre 1 y 6.

AC_23: Registrar ciudadano

Dado que: el ciudadano se encuentra en el formulario para registrarse.

Cuando: el ciudadano haga clic sobre el botón guardar

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de ciudadanos, el ciudadano registrado. El nuevo ciudadano deberá aparecer de primero en la lista.

4.1.12. Como usuario administrador necesita ver la lista de ciudadanos inscritos en una convocatoria para consultar la información

4.1.12.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar en una grilla los ciudadanos inscritos en una convocatoria.

El menú principal de la aplicación debe tener una opción que permita al usuario de la alcaldía abrir una página para ver el listado de los ciudadanos inscritos en una convocatoria.

En la página "ciudadanos inscritos" en la parte superior izquierda, debe haber una lista "Convocatorias" que permita al usuario administrador seleccionar la convocatoria para ver los ciudadanos inscritos.

La grilla ciudadanos inscritos deberá mostrar los siguientes datos:

- Tipo: tipo de identificación de ciudadano.
- Identificación: número de identificación del ciudadano.
- Primer nombre: primer nombre del ciudadano.

- Primer apellido: primer apellido del ciudadano.
- fecha de nacimiento: fecha de nacimiento del ciudadano.
- Estado civil: estado civil del ciudadano.
- Sisbén: puntaje del Sisbén del ciudadano.
- Ubicación: barrio o vereda de residencia del ciudadano.
- Zona: zona a la que pertenece el barrio o vereda
- Estado: pendiente, cumple requisitos, No cumple requisitos, información incompleta, aceptado.

La grilla "ciudadanos inscritos" debe mostrar a los ciudadanos inscritos, en orden descendente, por fecha de inscripción.

Si existen ciudadanos beneficiarios (estado aceptado) La grilla "ciudadanos inscritos" debe mostrarlos por defecto de primeros en orden descendente, por fecha de inscripción.

Cuando una convocatoria no tiene ciudadanos inscritos, la grilla "Ciudadanos Inscritos" deberá mostrar el encabezado sin ningún registro.

Ilustración 11 Prototipo listado ciudadanos inscritos

The screenshot shows a web browser window with the URL `http://www.timbio-cauca.gov.co/siseb`. The page title is "Ciudadanos Inscritos". Below the title, there is a "Convocatoria" dropdown menu. To the right, there is an "Acciones" menu with three options: "Inscribir Ciudadano", "Cerrar Convocatoria", and "Generar Beneficiarios". The main content area is a table with the following columns: Tipo, Identificación, Primer Nombre, Primer Apellido, Fecha Nacimiento, Estado Civil, Sisbén, Ubicación, Zona, and EstadoA. The table contains five rows of placeholder data. At the bottom of the page, there is a pagination bar showing "1-5 de 5 Elementos" and "20 Elementos por página".

Fuente: investigación propia

4.1.12.2. Criterios de Aceptación:

AC_01: Mostrar página ciudadanos inscritos

Dado que: el usuario administrador se encuentra en el menú principal de la aplicación.

Cuando: el usuario de clic en la opción "Ciudadanos Inscritos" del menú principal

Entonces:

- La aplicación deberá abrir la página "Ciudadanos Inscritos" que deberá mostrar en la parte superior izquierda una lista "convocatorias" con las convocatorias disponibles y por defecto debe estar en estado seleccionar.
- La grilla "Ciudadanos Inscritos" debe mostrar en el encabezado los campos "Tipo identificación", "Numero identificación", "Primer nombre", "Primer apellido", "Fecha nacimiento", "Estado civil", "Sisbén", "ubicación", "Zona", "estado" y no debe contener ningún registro.

AC_02: Mostrar ciudadanos inscritos en una convocatoria

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic en la lista "Convocatorias" y selecciona una de las convocatorias disponibles.

Entonces:

- La aplicación deberá mostrar en la grilla "ciudadanos inscritos" los ciudadanos inscritos en la convocatoria seleccionada.
- deben aparecer ordenados por fecha de inscripción en orden descendente.
- Si hay ciudadanos inscritos en estado "Aceptado" deben aparecer de primeros en la grilla organizados por fecha de inscripción en orden descendente.

AC_03: No existen ciudadanos inscritos

Dado que: el usuario administrador se encuentra en la página Ciudadanos inscritos.

Cuando: el usuario de clic en la lista "convocatorias", y seleccione una de las convocatorias disponibles.

Entonces: la grilla "ciudadanos inscritos" deberá mostrar el encabezado sin ningún registro.

4.1.13. Como ciudadano, necesita ver los resultados de la convocatoria para verificar si fue aceptado

4.1.13.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar al ciudadano los resultados de la convocatoria.

Cuando el ciudadano ingrese a la plataforma, esta debe mostrar en la pantalla de inicio los siguientes datos:

- Descripción: descripción de la convocatoria
- Nombre del proyecto: nombre del proyecto
- Fecha de inscripción: fecha de inscripción del ciudadano en una convocatoria
- convocatoria: estado de la convocatoria (abierta, cerrada).
- estado: estado del ciudadano en la convocatoria (aprobado(a), información incompleta).

Solo debe mostrar la información de las convocatorias en las que el ciudadano se haya inscrito.

La información debe aparecer al lado derecho de la pantalla de inicio en orden descendente por la fecha de inscripción.

4.1.13.2. Criterios de Aceptación:

AC_01: Consultar resultados convocatoria

Dado que: el ciudadano necesita consultar en la aplicación si salió beneficiado en una convocatoria.

Cuando: el ciudadano ingrese a la plataforma y se encuentre en la pantalla de inicio

Entonces:

- La aplicación deberá mostrar los datos "Descripción", "Nombre del proyecto", "Fecha de inscripción", "convocatoria", "estado".
- Solo debe mostrar la información de las convocatorias en las que el ciudadano se haya inscrito.
- Debe mostrar la información en orden descendente por "fecha de inscripción".

AC_04: El ciudadano no está inscrito en ninguna convocatoria

Dado que: el ciudadano no se encuentra inscrito en ninguna convocatoria.

Cuando: el ciudadano ingrese a la plataforma y se encuentre en la pantalla de inicio.

Entonces: no debe aparecer ningún tipo de información de las convocatorias.

4.1.14. Como usuario administrador, necesita buscar un ciudadano en la aplicación para consultar sus datos o actualizar la información

4.1.14.1. Descripción

El objetivo de esta historia de usuario consiste en buscar un ciudadano en la aplicación

La búsqueda de ciudadanos debe realizarse por el número de documento. Cuando el usuario ingrese un número de documento que exista en la base de datos, se deben mostrar la siguiente información:

- Tipo: tipo de identificación de ciudadano.
- Identificación: número de identificación del ciudadano.
- Primer nombre: primer nombre del ciudadano
- Segundo nombre: segundo nombre del ciudadano.
- Primer apellido: primer apellido del ciudadano.
- Segundo apellido: segundo apellido del ciudadano.

Ilustración 12 Prototipo buscar ciudadano

Fuente: investigación propia

4.1.14.2. Criterios de Aceptación:

AC_01: El Ciudadano ya se encuentra registrado en la aplicación

Dado que: el usuario administrador se encuentra en la página para buscar un ciudadano.

Cuando: el ciudadano ya se encuentra registrado y el administrador ingresa el número de documento y hace clic en el botón Buscar.

Entonces: La aplicación deberá mostrar la información del ciudadano "Tipo identificación", "Numero identificación", "Primer nombre", "Segundo nombre", "Primer apellido", "Segundo apellido".

AC_02: El ciudadano no existe

Dado que: el usuario administrador se encuentra en la página para buscar un ciudadano.

Cuando: el ciudadano NO se encuentra registrado y el administrador ingresa el número de documento y hace clic en el botón Buscar.

Entonces: La aplicación deberá mostrar un mensaje indicando que el ciudadano no se encuentra registrado "El ciudadano no se encuentra registrado en la aplicación".

4.1.15. Como usuario administrador necesito modificar la información de un ciudadano para mantener los datos actualizados.

4.1.15.1. Descripción

El objetivo de esta historia de usuario consiste en actualizar la información de un ciudadano para posteriormente asignarlo a una convocatoria.

La búsqueda de ciudadanos debe tener una opción para editar la información de los ciudadanos que se encuentran registrados en la aplicación.

Cuando el usuario haga clic en la opción para editar la información del ciudadano, la aplicación debe abrir la página para actualizar los datos.

Todos los campos del formulario deben mostrarse con la información del ciudadano a actualizar.

Un ciudadano debe tener la siguiente información:

- Tipo de identificación: debe mostrar un listado con los diferentes tipos de identificación "cedula de ciudadanía, cedula de extranjería, tarjeta de identidad, registro civil".
- Número de identificación: en caso de ser cambiado no puede existir más de un ciudadano con el mismo número de identificación y debe ser un entero positivo. El número de identificación debe contener letras y números sin espacios y caracteres especiales, y puede contener hasta 20 caracteres.
- Primer nombre: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Segundo nombre: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Primer apellido: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.
- Segundo apellido: este campo solo debe contener letras sin espacios y puede contener hasta 20 caracteres.

- Departamento de nacimiento: debe permitir seleccionar el departamento de Colombia en el que nació el ciudadano.
- Municipio de nacimiento: debe mostrar un listado de municipios que permita seleccionar el municipio en el que nació el ciudadano y que están relacionados al departamento de nacimiento seleccionado anteriormente.
- Fecha de nacimiento: la fecha de nacimiento debe ser menor o igual a la fecha actual
- Departamento de residencia: debe permitir seleccionar el departamento de Colombia en el que vive el ciudadano.
- Municipio de residencia: debe mostrar un listado de municipios que permita seleccionar el municipio en el que vive el ciudadano, y que están relacionados al departamento de residencia seleccionado.
- Zona de residencia: debe permitir seleccionar de una lista la zona de residencia de un ciudadano "urbana y rural".
- Estrato: número entero positivo entre 1 y 6.
- Barrio / Vereda de residencia: este campo debe permitir seleccionar la vereda o el barrio en el que vive el ciudadano y que esté relacionado con el municipio de residencia y la zona de residencia. Si anteriormente en el campo "Zona de residencia" el ciudadano selecciona "urbana", en este campo se deben mostrar los barrios del municipio y en caso contrario, si selecciona "rural" debe mostrar las veredas del municipio.
- Dirección: este campo puede contener cualquier tipo de carácter y máximo 30 caracteres.
- Teléfono Celular: este campo solo debe contener un número entero positivo.
- Email: este campo puede contener cualquier tipo de carácter, sin espacios y debe ser formato de correo electrónico.
- Estado civil: el estado civil debe tener las opciones Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a).
- Puntaje Sisbén: este campo debe contener un número positivo con máximo dos decimales y por defecto debe contener el valor de cero.
- Cabeza familia: SI o No.

Nota: los datos deben ser obligatorios a excepción del segundo nombre, segundo apellido, Departamento de nacimiento, municipio de nacimiento, dirección, email y teléfono celular.

Ilustración 13 Prototipo actualizar información del ciudadano

Siseb
http://www.timbio-cauca.gov.co/siseb

Buscar Ciudadanos

Ingrese el número de identificación del ciudadano:

.....

Información del ciudadano

Tipo de Identificación	Número de Identificación	Primer Nombre
.....
Segundo Nombre	Primer Apellido	Segundo Apellido
.....

Fuente: investigación propia

Ilustración 14

Siseb
http://www.timbio-cauca.gov.co/siseb

Registrar Ciudadano

Datos Personales

Tipo de Identificación	Número de Identificación	Primer Nombre
Seleccionar >>
Segundo Nombre	Primer Apellido	Segundo Apellido
.....
Estado Civil	Puntaje Sisbén	Cabeza Familia
Seleccionar >>	<input checked="" type="checkbox"/> SI

Datos Nacimiento

Departamento	Municipio	Fecha de Nacimiento
Seleccionar >>	Seleccionar >>

Datos Contacto

Departamento	Municipio	Zona
Seleccionar >>	Seleccionar >>	Seleccionar >>
Barrio / Vereda	Estrato	Dirección
Seleccionar >>
Teléfono Celular	Correo Electrónico	
.....	

Fuente: investigación propia

4.1.15.2. Criterios de Aceptación:

AC_01: Mostrar información del ciudadano a actualizar

Dado que: el usuario administrador se encuentra en la página de búsqueda de ciudadanos y el ciudadano se encuentra registrado en la aplicación.

Cuando: el usuario hace clic en la opción editar del ciudadano.

Entonces: La aplicación debe abrir el formulario de actualización de ciudadanos, mostrar la información del ciudadano y permitir su actualización.

AC_02: Número de identificación ya existe

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario cambia el número de identificación del ciudadano por otro que ya existe y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el número de identificación ya existe "El número de identificación ingresado ya se encuentra registrado en la aplicación."

AC_03: Campos obligatorios

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el campo es obligatorio y el usuario cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo el texto "Requerido".

AC_04: Fecha de nacimiento no puede ser mayor a la fecha actual

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario da clic en el campo fecha.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_05: El formato del correo electrónico no es el correcto

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario haya terminado de digitar el correo y si el formato es incorrecto y cambia el foco a otro campo.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el formato del correo electrónico es incorrecto "El formato del correo electrónico es incorrecto".

AC_06: Listado tipo de identificación

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario haga clic sobre la lista tipo de identificación.

Entonces: la aplicación deberá mostrar un listado de los tipos de identificación disponibles en la aplicación.

AC_07: Formato número de identificación

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el número de identificación.

Entonces: el campo debe restringir cualquier carácter excepto letras y números y solo debe permitir 20 caracteres.

AC_08: Formato primer nombre

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el primer nombre del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_09: Formato segundo nombre

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el segundo nombre del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_10: Formato primer apellido

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el primer apellido del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_11: Formato segundo apellido

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el segundo apellido del ciudadano.

Entonces: el campo debe restringir cualquier carácter excepto letras y debe permitir máximo 20 caracteres.

AC_12: Listado departamentos de nacimiento

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_13: Listado municipios de nacimiento

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de nacimiento seleccionado anteriormente.

AC_14: Formato fecha de nacimiento

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre el campo fecha de nacimiento.

Entonces: la aplicación debe mostrar el calendario activo solo hasta la fecha actual.

AC_15: Listado departamentos de residencia

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista departamentos.

Entonces: la aplicación deberá mostrar un listado de los departamentos de Colombia disponibles en la base de datos.

AC_16: Listado municipios de residencia

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista municipios.

Entonces: la aplicación deberá mostrar un listado de los municipios de Colombia disponibles en la base de datos y que estén relacionados al departamento de residencia seleccionado anteriormente.

AC_17: Listado zonas

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista zonas.

Entonces: la aplicación deberá mostrar una lista con las opciones "urbana, rural".

AC_18: Listado barrios/veredas de residencia

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario de clic sobre la lista barrios/veredas.

Entonces:

- La aplicación deberá mostrar un listado de los barrios o veredas disponibles en la base de datos y que estén relacionados al departamento de residencia y a la zona seleccionados anteriormente.
- Si la zona de residencia seleccionada anteriormente es urbana debe mostrar los barrios del municipio en caso contrario cuando es rural debe mostrar las veredas.

AC_19: Formato número de teléfono

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el número de teléfono.

Entonces: el campo debe restringir cualquier carácter excepto números.

AC_20: Formato Email

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario digite el correo electrónico.

Entonces: el campo debe permitir cualquier carácter excepto espacios y validar que este escrito en formato de correo electrónico.

AC_21: Campo estado civil

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de clic en la lista "estado civil"

Entonces: la aplicación debe mostrar las opciones "Casado(a), Soltero(a), Unión Libre, divorciado(a), viudo(a)".

AC_22: Campo puntaje Sisbén

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el puntaje del Sisbén

Entonces: el campo debe permitir solo números positivos con máximo dos decimales

AC_23: Campo estrato

Dado que: el usuario administrador se encuentra en el formulario para registrar un ciudadano.

Cuando: el usuario de digite el estrato

Entonces: el campo debe permitir solo números positivos entre 1 y 6.

AC_24: Actualizar ciudadano

Dado que: el usuario de planeación se encuentra en el formulario para actualizar los datos de un ciudadano.

Cuando: el usuario haga clic sobre el botón guardar.

Entonces: la aplicación debe cerrar el formulario y mostrar en la lista de ciudadanos, la información del ciudadano actualizada.

4.1.16. Como usuario administrador, necesita ver los beneficiarios de un proyecto para consultar la información.

4.1.16.1. Descripción

El objetivo de esta historia de usuario consiste en mostrar en una grilla los beneficiarios de un proyecto.

La grilla deberá mostrar los siguientes datos:

- Tipo: tipo de identificación del ciudadano.
- Identificación: número de identificación del ciudadano.

- Primer nombre: primer nombre del ciudadano
- Segundo nombre: segundo nombre del ciudadano.
- Primer apellido: primer apellido del ciudadano.
- Segundo apellido: segundo apellido del ciudadano.
- Ubicación: barrio o vereda de residencia del ciudadano.

Los ciudadanos deben mostrarse ordenados por el primer apellido en orden ascendente.

Ilustración 15 Prototipo listado beneficiarios

The screenshot shows a web browser window with the URL `http://www.timbio-cauca.gov.co/siseb`. The page title is "Beneficiarios". Below the title, there is a "Proyecto:" label followed by a dashed line. The main content is a table with the following columns: "Tipo", "Identificación", "Primer Nombre", "Segundo Nombre", "Primer Apellido A", "Segundo Apellido", and "Ubicación". The table contains five rows of placeholder data, each row starting with a small square icon. At the bottom of the page, there is a pagination control showing "1-5 de 5 Elementos", a page number "1" in a box, and "20 Elementos por página".

Fuente: investigación propia

4.1.16.2. Criterios de Aceptación

AC_01: Existen beneficiarios

Dado que: el usuario administrador se encuentra en la lista de proyectos y necesita ver los beneficiarios de un proyecto.

Cuando: el usuario hace clic en la opción "beneficiarios" de un Proyecto y el proyecto ya tiene beneficiarios.

Entonces:

- La aplicación deberá mostrar en una grilla los ciudadanos beneficiarios de un proyecto.
- La información que se debe mostrar por cada ciudadano beneficiario debe ser "Tipo", "Identificación", "Primer nombre", "Segundo nombre", "Primer apellido", "Segundo apellido", "Ubicación".
- La información de los ciudadanos beneficiarios debe mostrarse ordenada por el Primer Apellido en orden ascendente.

AC_02: No existen beneficiarios

Dado que: el usuario administrador se encuentra en la lista de proyectos y necesita ver los beneficiarios de un proyecto.

Cuando: el usuario hace clic en la opción "Beneficiarios" de un proyecto y el proyecto NO tiene beneficiarios.

Entonces: La aplicación deberá mostrar el texto "Sin registros" debajo de los nombres de las columnas de la grilla.

4.1.17. Como usuario administrador, necesita revisar los requisitos de un ciudadano inscrito en una convocatoria para validar la información

4.1.17.1. Descripción

El objetivo de esta historia de usuario consiste en verificar los requisitos de un ciudadano inscrito en una convocatoria.

La parte superior izquierda de la página, debe tener una lista "Convocatorias" que permita al usuario administrador seleccionar una de las convocatorias, para ver los ciudadanos inscritos.

Cada fila de la grilla "ciudadanos inscritos" debe tener una opción que muestre el formulario "Ciudadanos" con los datos del ciudadano inscrito en la convocatoria y los requisitos (documentos) configurados en la convocatoria.

El formulario debe contener los siguientes datos.

- Identificación: número de identificación del ciudadano.
- Primer Nombre: primer nombre del ciudadano.
- Segundo Nombre: segundo nombre del ciudadano.
- Primer Apellido: primer apellido del ciudadano.
- Segundo Apellido: segundo apellido del ciudadano.
- Fecha de Nacimiento: fecha de nacimiento del ciudadano.
- Zona: zona de residencia del ciudadano.
- Estado civil: estado civil del ciudadano.
- Puntaje Sisbén: puntaje del Sisbén del ciudadano.
- Estado: estado del ciudadano en la convocatoria (pendiente, cumple requisitos, No cumple requisitos, información incompleta, aceptado).
Cada requisito(documento) debe tener al lado derecho las siguientes opciones.
- Cargar el documento.
- Mostrar el documento.
- Chequear requisito si cumple.

Debe haber un botón que permita guardar los cambios efectuados por el usuario administrador.

el estado del ciudadano debe cambiar de acuerdo a los cambios efectuados por el usuario administrador.

4.1.17.2. Criterios de Aceptación:

AC_01: Lista convocatorias

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic sobre la lista de "Convocatorias".

Entonces: la aplicación deberá mostrar un listado de las convocatorias almacenados en la base de datos.

AC_02: Mostrar ciudadanos inscritos

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic en la lista "Convocatorias" y seleccione una de las convocatorias disponibles.

Entonces: la aplicación deberá mostrar en la grilla "Ciudadanos Inscritos" los ciudadanos inscritos en la convocatoria seleccionada.

AC_03: Mostrar información del ciudadano y requisitos exigidos por convocatoria

Dado que: el usuario administrador se encuentra en el listado de ciudadanos inscritos y necesita verificar los requisitos de un ciudadano(a) inscrito en una convocatoria.

Cuando: el usuario haga clic en la opción "explorar" de un ciudadano inscrito.

Entonces: La aplicación debe mostrar en un formulario los datos del ciudadano inscrito (Identificación, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Fecha de Nacimiento, Zona, Estado civil, Puntaje Sisbén, Estado) y los requisitos (documentos) exigidos por cada convocatoria.

AC_04: Cargar documentos

Dado que: el usuario administrador se encuentra en la página con los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en la opción "Adjuntar" de un requisito.

Entonces: la aplicación debe mostrar una ventana con el explorador de Windows que permita seleccionar el archivo almacenado en el dispositivo.

AC_05: Ver documentos

Dado que: el usuario administrador se encuentra en la página con los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en la opción "Ver" de un requisito.

Entonces: la aplicación debe mostrar el archivo adjunto.

AC_06: Validar requisito

Dado que: el usuario administrador se encuentra en la página con los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en la caja de verificación de un requisito.

Entonces: la aplicación deberá mostrar un icono en color verde que indique que el requisito se encuentra validado

AC_07: Guardar cambios

Dado que: el usuario administrador se encuentra en la página con la información de un ciudadano inscrito y los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en el botón Guardar

Entonces: la aplicación debe guardar los cambios efectuados por el usuario administrador y actualizar el estado (pendiente, cumple requisitos, No cumple requisitos, información incompleta, aceptado).

4.1.18. Como usuario administrador, necesita inscribir un ciudadano en una convocatoria para que pueda participar en la selección de beneficiarios para un proyecto

4.1.18.1. Descripción

El objetivo de esta historia de usuario consiste en registrar un ciudadano en una convocatoria.

El menú principal de la aplicación debe tener una opción que permita al usuario de la alcaldía abrir una página (Ciudadanos Inscritos) para inscribir un ciudadano en una convocatoria.

La página "Ciudadanos Inscritos" Debe tener en la parte superior derecha una opción para inscribir un ciudadano en una convocatoria. La opción debe mostrarse en estado deshabilitado.

La parte superior izquierda de la página, debe tener una lista "Convocatorias" que permita al usuario administrador seleccionar una convocatoria para inscribir un ciudadano(a). Cuando el usuario administrador seleccione la convocatoria, el botón "inscribir ciudadano(a)" quedará habilitado.

cuando el usuario administrador de clic en el botón "inscribir ciudadano" la aplicación debe abrir una página que muestre el formulario "ciudadano" y los requisitos configurados en la convocatoria.

El formulario ciudadano debe estar vacío y contener los siguientes campos.

- Identificación: número de identificación del ciudadano.
- Primer Nombre: primer nombre del ciudadano.
- Segundo Nombre: segundo nombre del ciudadano.
- Primer Apellido: primer apellido del ciudadano.
- Segundo Apellido: segundo apellido del ciudadano.
- Fecha de Nacimiento: fecha de nacimiento del ciudadano.
- Zona: zona de residencia del ciudadano.
- Estado civil: estado civil del ciudadano.
- Puntaje Sisbén: puntaje del Sisbén del ciudadano.

El campo "Identificación" debe permitir digitar el número de identificación del ciudadano para poder buscarlo e inscribirlo en una convocatoria.

Cuando el ciudadano no se encuentra registrado la aplicación debe permitir abrir el formulario ciudadano para poder registrarlo.

Cada requisito(documento) debe tener al lado derecho las siguientes opciones

- Cargar el documento
- Mostrar el documento
- Chequear los requisitos que si cumplen.

Debe mostrar el estado en que se encuentra el ciudadano en la convocatoria.

Debe mostrar un botón que permita inscribir el ciudadano en la convocatoria.

Ilustración 16 Prototipo Inscribir ciudadano en una convocatoria

Fuente: investigación propia

4.1.18.2. Criterios de Aceptación:

AC_01: Mostrar página ciudadanos inscritos

Dado que: el usuario administrador se encuentra en el menú principal de la aplicación.

Cuando: el usuario de clic en la opción "Ciudadanos Inscritos" del menú principal

Entonces:

- La aplicación deberá abrir la página "Ciudadanos Inscritos" que deberá mostrar en la parte superior izquierda una lista "convocatorias" con las convocatorias disponibles y por defecto debe estar en estado seleccionar.
- La grilla "Ciudadanos Inscritos" debe mostrar en el encabezado los campos "Tipo identificación", "Numero identificación", "Primer nombre", "Primer apellido", "Fecha nacimiento", "Estado civil", "Sisbén", "ubicación", "Zona", "estado" y no debe contener ningún registro.
- debe mostrar una opción (Inscribir ciudadano) en estado deshabilitado.

AC_02: Listado convocatorias

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic sobre la lista de "Convocatorias".

Entonces: la aplicación deberá mostrar un listado de las convocatorias almacenados en la base de datos.

AC_03: Habilitar opción "Inscribir Ciudadano"

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic en la lista "Convocatorias" y seleccione una de las convocatorias disponibles.

Entonces: La aplicación deberá habilitar la opción "Inscribir Ciudadano"

AC_04: Mostrar formulario Ciudadanos y requisitos configurados por convocatoria.

Dado que: el usuario administrador se encuentra en la página "Ciudadanos Inscritos"

Cuando: el usuario haga clic en la opción "Inscribir Ciudadano"

Entonces: La aplicación deberá mostrar en la página "Inscribir Ciudadano", el formulario "Ciudadanos" con los campos vacíos y los requisitos configurados por convocatoria.

AC_05: El ciudadano ya está inscrito en una convocatoria

Dado que: el usuario administrador se encuentra en la página "Inscribir Ciudadano"

Cuando: el usuario digita en el campo "Identificación" el un número de identificación de un ciudadano, y da clic en el botón Buscar.

Entonces: la aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el ciudadano ya se encuentra registrado en la convocatoria "El ciudadano ya está inscrito en la convocatoria".

AC_06: El ciudadano no se encuentra registrado en la aplicación

Dado que: el usuario administrador se encuentra en la página "Inscribir Ciudadano"

Cuando: el usuario ingresa el número de identificación de un ciudadano, y da clic en el botón Buscar.

Entonces:

- La aplicación debe mostrar debajo del campo, un mensaje indicándole al usuario que el ciudadano no se encuentra registrado "el ciudadano no se encuentra registrado, ¿desea ingresarlo?".
- El mensaje de validación debe contener un botón "Aceptar" que habrá el formulario "Ciudadanos" para poder registrarlo y otro botón (Cancelar) que permita cancelar la operación.

AC_07: Mostrar datos del ciudadano

Dado que: el usuario administrador se encuentra en la página "Inscribir Ciudadano".

Cuando: el usuario digite en el campo "Identificación" el número de identificación del ciudadano y de clic en el botón "Buscar"

Entonces: la aplicación deberá mostrar en el formulario los datos del ciudadano (Identificación, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Fecha de Nacimiento, Zona, Estado civil, Puntaje Sisbén).

AC_08: Cargar documentos de un ciudadano

Dado que: el usuario administrador se encuentra en la página "Inscribir Ciudadano"

Cuando: el usuario de clic en la opción "Adjuntar" de un requisito, de un requisito.

Entonces: la aplicación debe mostrar una ventana con el explorador de Windows que permita seleccionar el archivo almacenado en el dispositivo.

AC_09: Ver documentos de un ciudadano

Dado que: el usuario administrador se encuentra en la página "Inscribir Ciudadano".

Cuando: el usuario de clic en la opción "Ver" de un requisito.

Entonces: la aplicación debe mostrar el archivo adjunto.

AC_10: Validar requisito

Dado que: el usuario administrador se encuentra en la página “Inscribir Ciudadano”.

Cuando: el usuario de clic en la caja de verificación de un requisito.

Entonces: la aplicación deberá mostrar un icono en color verde que indique que el requisito se encuentra validado

AC_11: Inscribir ciudadano en una convocatoria

Dado que: el usuario administrador se encuentra en la página “Inscribir Ciudadano”.

Cuando: el usuario de clic en el botón "Guardar"

Entonces:

- La aplicación deberá mostrar un mensaje informándole al usuario administrador que el ciudadano(a) se ha inscrito satisfactoriamente a la convocatoria "El ciudadano ha sido inscrito satisfactoriamente a la convocatoria.
- Deberá cerrar la página (inscribir ciudadano en una convocatoria) y mostrar en la lista "ciudadanos inscritos", el ciudadano inscrito en la convocatoria por fecha de inscripción en orden descendente.

4.1.19. Como ciudadano necesita registrarse en una convocatoria para poder participar en la selección de beneficiarios para un proyecto.

4.1.19.1. Descripción

El objetivo de esta historia de usuario consiste en que un ciudadano se pueda inscribir en una convocatoria.

En la pantalla de inicio de la aplicación, en la parte superior derecha debe haber un botón que permita al ciudadano inscribirse en una convocatoria.

Cuando el ciudadano de clic en el botón "inscribirse en convocatoria" la aplicación debe abrir una página que contenga una lista "Convocatorias", que permita al ciudadano(a) seleccionar una convocatoria para poder inscribirse.

Cuando el usuario selecciona una convocatoria de la lista "Convocatorias" la aplicación deberá mostrar los datos de la convocatoria

Los datos que se deben mostrar de la convocatoria son los siguientes:

- Proyecto: nombre del proyecto.
- Fecha de Apertura: fecha de aprobación.
- Fecha de Cierre: fecha de cierre
- Descripción: objetivo descripción de la convocatoria.
- Estado: puede contener los valores "abierto, cerrado, inactivo"

Debe existir un botón que permita al ciudadano inscribirse en la convocatoria y una opción que abra el formulario "Ciudadanos" si el ciudadano no se ha registrado en la aplicación.

Ilustración 17 Prototipo inscribirse en convocatoria

El prototipo muestra una interfaz de usuario en un navegador web. La barra de direcciones muestra la URL <http://www.timbio-cauca.gov.co/siseb>. El título de la página es "Inscribirse en Convocatoria". En la parte superior derecha, hay un campo de texto etiquetado "Nombre del Ciudadano".

El formulario principal contiene los siguientes elementos:

- Un menú desplegable etiquetado "Convocatoria:" con el texto "Seleccionar >>" y una flecha hacia abajo.
- Una línea horizontal azul que separa la sección "Convocatoria" de los campos de entrada.
- Un campo de texto etiquetado "Proyecto".
- Dois campos de texto etiquetados "Fecha de Apertura" y "Fecha de Cierre".
- Un campo de texto etiquetado "Descripción".
- Un campo de texto etiquetado "Estado".
- Dois botones en la parte inferior derecha: "Cancelar" y "Guardar".

Fuente: investigación propia

4.1.19.2. Criterios de Aceptación:

AC_01: Mostrar página "Inscribirse en Convocatoria"

Dado que: el ciudadano se encuentra en la pantalla de inicio de la aplicación.

Cuando: el ciudadano de clic en el botón "Inscribirse en Convocatoria".

Entonces:

- La aplicación deberá abrir la página "Inscribirse en Convocatoria" que deberá mostrar en la parte superior izquierda una lista "convocatorias" con las convocatorias disponibles y por defecto debe estar en estado seleccionar.
- debe mostrar un botón (Inscribirse) en estado deshabilitado.

AC_02: Lista convocatorias

Dado que: el ciudadano se encuentra en la página "Inscribirse en Convocatoria"

Cuando: el ciudadano haga clic sobre la lista de "Convocatorias".

Entonces: la aplicación deberá mostrar un listado de las convocatorias almacenados en la base de datos.

AC_03: Mostrar información de la convocatoria

Dado que: el ciudadano se encuentra en la página "Inscribirse en Convocatoria"

Cuando: el ciudadano haga clic en la lista "Convocatorias" y seleccione una de las convocatorias disponibles.

Entonces: La aplicación deberá mostrar los datos de la convocatoria. (Proyecto, Fecha de Apertura, Fecha de Cierre, Descripción, Estado)

AC_05: El ciudadano ya está inscrito en una convocatoria

Dado que: el ciudadano se encuentra en la página "Inscribirse en Convocatoria".

Cuando: el ciudadano de clic en el botón "Inscribirse".

Entonces: la aplicación deberá mostrar un mensaje indicándole al ciudadano que ya se encuentra inscrito en la convocatoria "El ciudadano ya está inscrito en la convocatoria".

AC_06: El ciudadano no se encuentra registrado en la aplicación

Dado que: el ciudadano se encuentra en la página "Inscribirse en Convocatoria".

Cuando: el ciudadano de clic en el botón "Inscribirse".

Entonces:

- La aplicación deberá mostrar un mensaje indicándole al ciudadano que no se encuentra registrado "El ciudadano no se encuentra registrado ¿desea hacerlo?"
- El mensaje debe contener un botón "Aceptar" que habra el formulario "Ciudadanos" para que el ciudadano pueda registrarse y otro botón (Cancelar) que permita cancelar la operación.

AC_11: Inscribirse en una convocatoria

Dado que: el ciudadano se encuentra en la página "Inscribirse en Convocatoria".

Cuando: el usuario de clic en el botón "Inscribirse"

Entonces:

- La aplicación deberá mostrar un mensaje informándole al ciudadano que la inscripción fue exitosa "su inscripción ha sido exitosa"
- Deberá cerrar la página (inscribirse en convocatoria) y mostrar en la pantalla de inicio de la plataforma el ciudadano inscrito en la convocatoria por fecha de inscripción en orden descendente.

4.1.20. Como ciudadano, necesita cargar los documentos para poder cumplir con los requisitos exigidos por la convocatoria

4.1.20.1. Descripción

El objetivo de esta historia de usuario, consiste en que un ciudadano pueda cargar los documentos exigidos en una convocatoria.

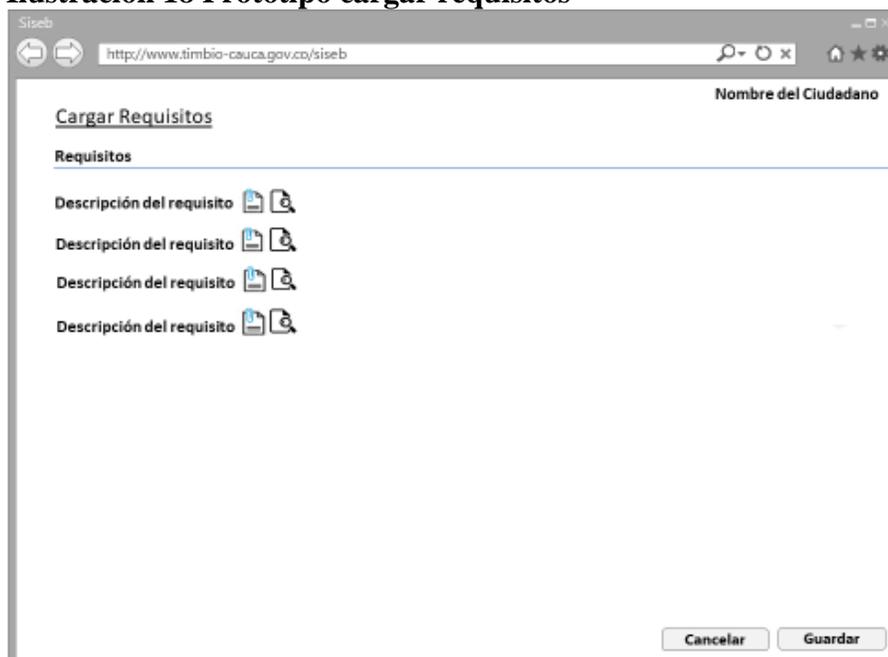
En la pantalla de inicio de la aplicación, en cada una de las inscripciones realizadas por el ciudadano, debe haber una opción que permita abrir una página para cargar los documentos exigidos en una convocatoria.

Cada requisito(documento) debe tener al lado derecho una opción para cargar el archivo y otra para ver el archivo cargado.

Debe haber un botón que permita guardar los cambios efectuados por el ciudadano.

El estado del ciudadano en la convocatoria debe cambiar de acuerdo a los cambios efectuados por el ciudadano.

Ilustración 18 Prototipo cargar requisitos



Fuente: investigación propia

4.1.20.2. Criterios de Aceptación:

AC_04: Abrir página "cargar documentos Convocatoria"

Dado que: el ciudadano se encuentra en la pantalla de inicio con la información de cada una de las inscripciones en una convocatoria.

Cuando: el ciudadano de clic en la opción "Cargar Documentos" de una inscripción.

Entonces: la aplicación debe abrir la página "cargar documentos Convocatoria" con los requisitos configurados en una convocatoria.

AC_04: Cargar documentos

Dado que: el ciudadano se encuentra en la página con los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en la opción "Adjuntar" de un requisito.

Entonces: la aplicación debe mostrar una ventana con el explorador de Windows que permita seleccionar el archivo almacenado en el dispositivo.

AC_05: Ver documentos

Dado que: el ciudadano se encuentra en la página con los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en la opción "Ver" de un requisito.

Entonces: la aplicación debe mostrar el archivo adjunto.

AC_07: Guardar cambios

Dado que: el ciudadano se encuentra en la página con la información de un ciudadano inscrito y los requisitos exigidos por convocatoria.

Cuando: el usuario de clic en el botón Guardar.

Entonces: la aplicación debe guardar los cambios efectuados por el usuario administrador y actualizar el estado (pendiente, cumple requisitos, No cumple requisitos, información incompleta, aceptado).

4.1.21. Como usuario administrador, necesita generar los beneficiarios de una convocatoria para asignarlos a un proyecto.

4.1.21.1. Descripción

El objetivo de esta historia de usuario consiste en generar los beneficiarios de una convocatoria.

La página "Ciudadanos Inscritos" debe contener una opción que permita generar los beneficiarios de una convocatoria. Esta opción por defecto debe estar deshabilitada y habilitarse cuando el usuario seleccione un elemento de la lista de convocatorias.

Cuando el usuario haga clic sobre la opción "Generar Beneficiarios", la aplicación debe mostrar un mensaje solicitándole al usuario que confirme la acción "Presione aceptar para confirmar la generación de beneficiarios de la convocatoria."

El proceso de generación de beneficiarios está compuesto por los siguientes puntos:

Por cada ciudadano inscrito en la convocatoria, La aplicación debe verificar si el ciudadano ha cargado los documentos exigidos como requisito en la convocatoria.

- a. Si los documentos están completos, la aplicación debe verificar si el ciudadano cumple con los requisitos automáticos, es decir requisitos diferentes al tipo de requisito "Documento", los cuales pueden ser verificados con la información del ciudadano almacenada en la aplicación.
- b. Si el ciudadano cumple con los requisitos automáticos, validar si todos los requisitos de tipo documento se encuentran verificados.
 - I. Si los requisitos se encuentran verificados, la aplicación debe actualizar el estado del beneficiario en la convocatoria a estado "Aprobado".
 - II. Si los requisitos no se encuentran verificados, la aplicación debe actualizar el estado del beneficiario en la convocatoria a estado "Pendiente verificación documentos".
- c. Si el ciudadano NO cumple con los requisitos automáticos, la aplicación debe actualizar el estado del beneficiario en la convocatoria a estado "No cumple requisitos".
- d. Si los documentos no están completos, la aplicación debe actualizar el estado del ciudadano en la convocatoria a estado "Documentación Incompleta".

4.1.21.2. Criterios de Aceptación:

AC_01: Documentación incompleta

Dado que: el sistema se encuentra generando los beneficiarios de una convocatoria.

Cuando: un ciudadano no ha cargado los documentos exigidos por la convocatoria.

Entonces: el sistema debe actualizar el estado del ciudadano en la convocatoria a estado "Documentación Incompleta".

AC_02: El ciudadano no cumple con los requisitos básicos.

Dado que: el sistema se encuentra generando los beneficiarios de una convocatoria.

Cuando: el ciudadano ha cargado todos los documentos exigidos por la convocatoria, pero no cumple con los requisitos diferentes a los requisitos de tipo documento.

Entonces: el sistema debe actualizar el estado del ciudadano en la convocatoria a estado "No cumple requisitos".

AC_03: Los documentos pendientes de verificación.

Dado que: el sistema se encuentra generando los beneficiarios de una convocatoria.

Cuando: el ciudadano ha cargado todos los documentos exigidos por la convocatoria, cumple con los requisitos diferentes a los requisitos de tipo documento y tiene documentos pendientes de verificación.

Entonces: el sistema debe actualizar el estado del ciudadano en la convocatoria a estado "Pendiente verificación documentos".

AC_04: Ciudadano aprobado

Dado que: el sistema se encuentra generando los beneficiarios de una convocatoria.

Cuando: el ciudadano ha cargado todos los documentos exigidos por la convocatoria, cumple con los requisitos diferentes a los requisitos de tipo documento y los documentos cargados se encuentran verificados.

Entonces: el sistema debe actualizar el estado del ciudadano en la convocatoria a estado "Aprobado".

4.1.23. Portabilidad

La aplicación debe ser portable y permitir la instalación y ejecución en sistemas operativos Microsoft Windows, macOS y Linux.

4.1.24. Usabilidad

A continuación, se listan los requerimientos de usabilidad que debe cumplir la aplicación.

- Todas las grillas deben tener paginación y por defecto se deben mostrar 20 registros.
- Todas las columnas de las grillas deben permitir filtrar la información.
- Todas las columnas de las grillas deben tener la opción para ordenar los registros ascendentemente o descendentemente.
- Las listas de ítems deben mostrar por defecto la opción "Seleccionar >>" cuando el formulario es para un nuevo registro.
- Las listas de ítems deben permitir que el usuario filtre la información, digitando un texto completo o parte del texto. La lista se debe filtrar a medida que el usuario digita el texto.
- Cuando los campos de un formulario son obligatorios y el usuario no selecciona o digita información y cambia el foco a otro campo, se debe mostrar debajo del campo, el texto "Requerido".

4.1.25. Auditoria

A continuación, se listan los requerimientos de auditoria que debe cumplir la aplicación.

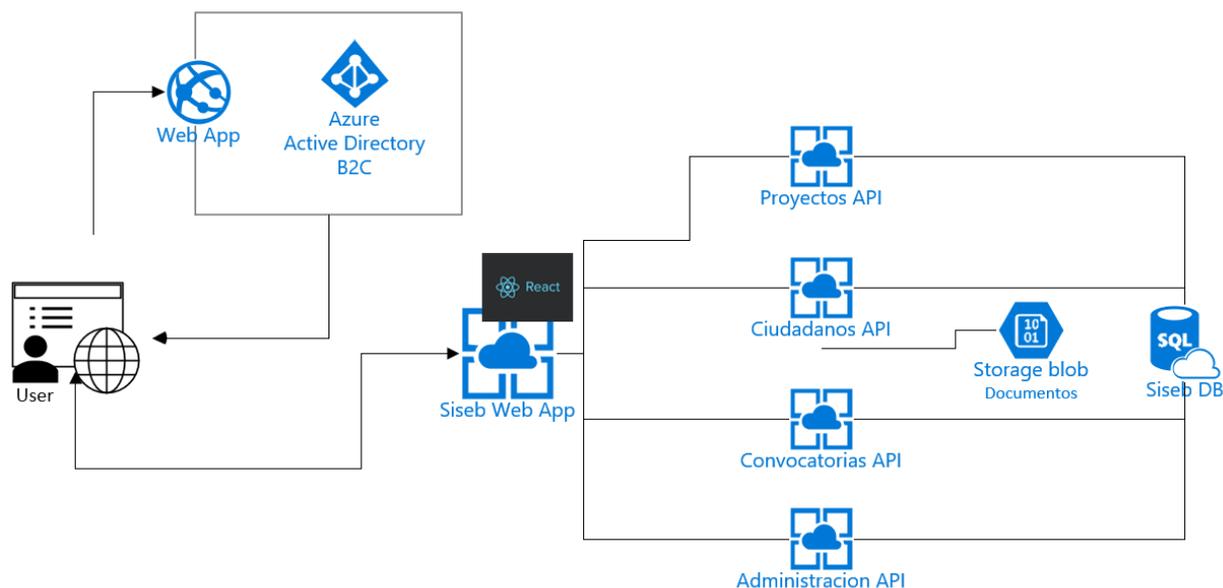
- Todas las tablas de la base de datos deben tener los campos: fecha de creación, fecha de última actualización, alias del usuario quien crea el registro, alias del usuario quien realiza la última actualización.
- No está dentro del alcance del proyecto, la auditoria de los cambios realizados por cada campo de una tabla.

4.2. ARQUITECTURA DE LA APLICACIÓN

La solución se desplegará completamente en PaaS (Platform as a service) en la nube de Microsoft Azure. PaaS les permite a las organizaciones evitar gastos en hardware y licenciamiento, proporcionando recursos como bases de datos, servicios de almacenamiento de archivos, alojamiento para aplicaciones Web y APIs, servicios de autenticación y autorización, de tal manera que las empresas puedan tener sus ambientes de desarrollo, pruebas y producción en la nube,

pagando solo por los recursos utilizados. A continuación, se muestra el diagrama con la arquitectura propuesta y la descripción de los recursos que la componen.

Ilustración 19 Diagrama Arquitectura de la Aplicación



Fuente: investigación propia

En el diagrama anterior, se puede observar que un usuario intenta ingresar a la aplicación a través de un navegador, si el usuario no se encuentra autenticado, la aplicación redirecciona al usuario a una página para seleccionar si se va a autenticar con su cuenta corporativa o con su cuenta de Facebook. Si la autenticación es con una cuenta corporativa (Funcionarios Alcaldía), entonces la aplicación solicita ingresar el usuario y la clave, si la autenticación es con Facebook (Ciudadanos), la aplicación redirecciona al usuario a la página de autenticación de Facebook. El proceso de autenticación con una cuenta corporativa se realiza a través del Azure Active Directory y el proceso con la cuenta de Facebook a través del Azure Active Directory B2C.

Cuando el proceso de autenticación es exitoso, se redirecciona al usuario a la página de bienvenida de la aplicación Web Siseb. La página de bienvenida es diferente dependiendo del rol del usuario (Ciudadano o Administrador).

La aplicación Web consume REST APIs (.NET Core) para enviar y recibir información de la base de datos y para implementar la lógica de negocio de la aplicación. Adicional, el API de beneficiarios utiliza un Storage blob para almacenar los documentos que cargan los usuarios.

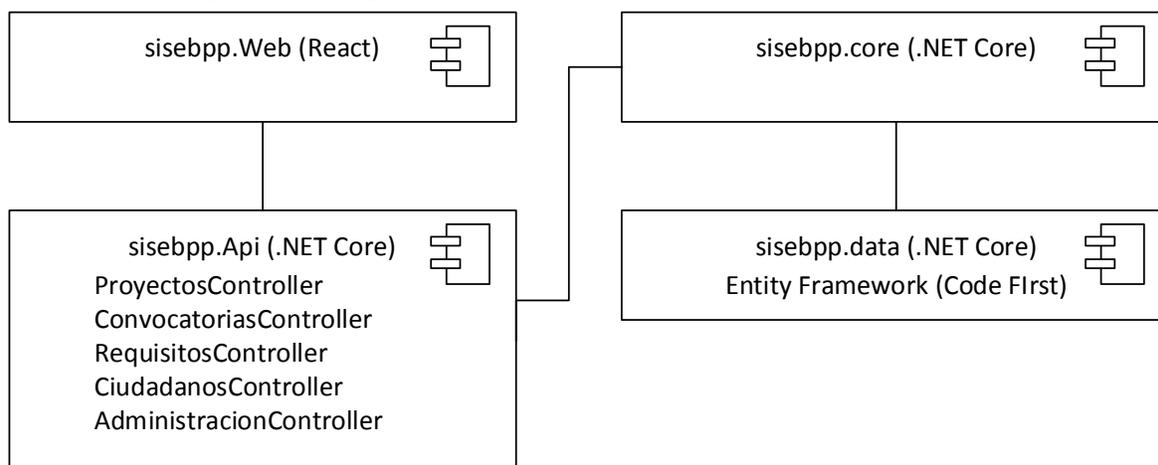
Tabla 2 Arquitectura de la aplicación

Recurso	Descripción
Azure Active Directory B2C	Azure Active Directory B2C proporciona business-to-customer identity como servicio y permite usar cuentas de diferentes redes sociales, para obtener un único acceso de inicio de sesión a aplicaciones y APIs. La aplicación usará la autenticación del proveedor Facebook para permitir el acceso de los ciudadanos a la aplicación.
Sisebpp Web App	Servicio de Azure basado en HTTP para alojar la aplicación Web de la aplicación. Desarrollado por el estudiante.
Proyectos API	Servicio de Azure basado en HTTP para alojar la API de proyectos. Desarrollado por el estudiante.
Convocatorias API	Servicio de Azure basado en HTTP para alojar la API de convocatorias. Desarrollado por el estudiante.
Ciudadanos API	Servicio de Azure basado en HTTP para alojar la API de ciudadanos. Desarrollado por el estudiante.
Administración API	Servicio de Azure basado en HTTP para alojar la API de administración. Desarrollado por el estudiante.
Storage blob Documentos	Blob storage de Azure para almacenar los documentos de requisitos cargados por los ciudadanos.
Sisebpp DB	Base de datos en SQL Azure para almacenar y administrar la información de la aplicación.

Fuente: investigación propia

4.3. COMPONENTES DE DESARROLLO

Ilustración 20 diagrama Componentes de Desarrollo



Fuente: investigación propia

Como se muestra en la imagen anterior, el desarrollo de la aplicación tiene 4 componentes, el componente de la aplicación Web (sisebpp.Web) que consume las APIs implementadas en el componente sisebpp.Api, las cuales se comunican con el componente sisebpp.core, donde se encuentra la implementación de la lógica de negocio y se envían las peticiones al componente de datos (sisebpp.data).

Tabla 3 Componentes de Desarrollo

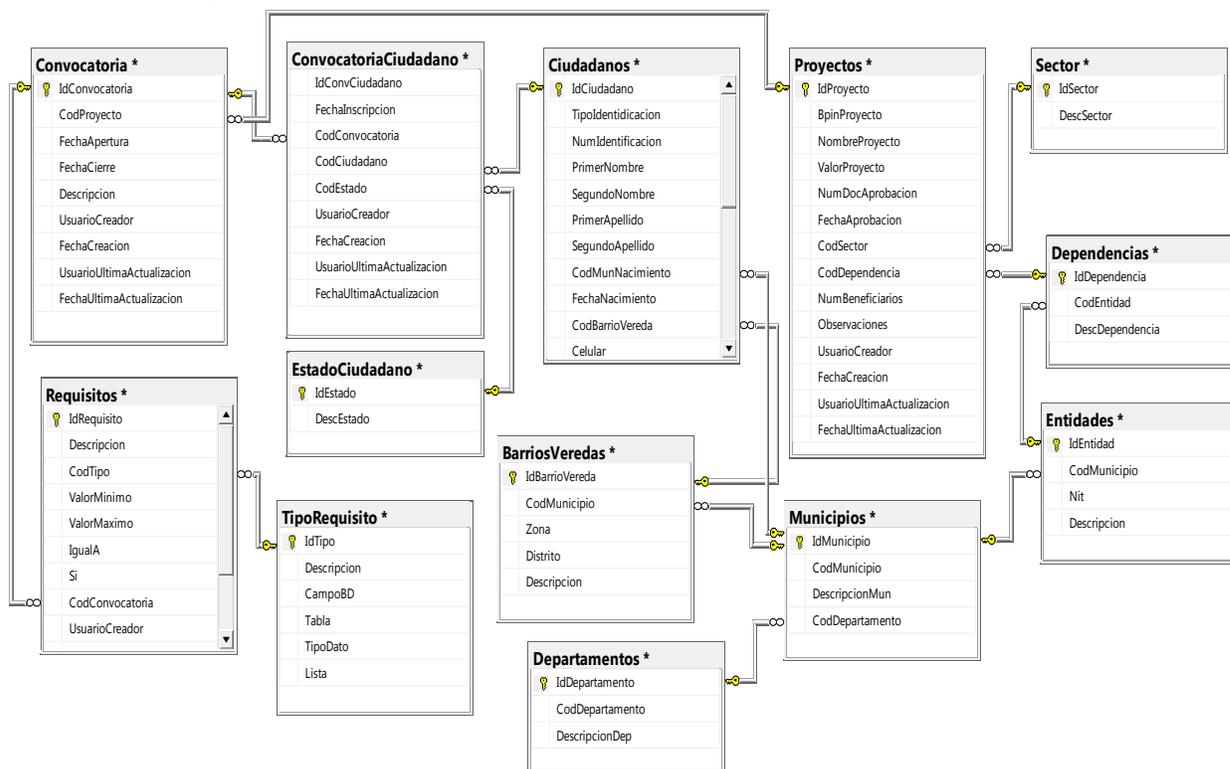
Componente	Descripción
sisebpp.Web	La implementación de la aplicación Web se está desarrollando con la librería de Front-End React y el patron SPA (Single Page Application), la cual permite la creación de interfaces de usuario o componentes a través de JavaScript.
sisebpp.Api	Componente .NET Core, programado en C#, el cual contiene los controladores de las APIs de Proyectos, Convocatorias, Requisitos, Ciudadanos y Administración.
sisebpp.core	Componente .NET Core, programado en C#, el cual contiene los modelos de las APIs, las clases con la lógica de negocio y los métodos para enviar y recibir información de la capa de datos.
sisebpp.data	Componente .NET Core, programado en C#, el cual usa el ORM (Object Relational Mapper) Entity Framework Core para construir la capa de acceso a la base de datos SQL Server. La

implementación de este componente usa el patrón Code-First, que permite crear las tablas de la base de datos, sus relaciones y constraints utilizando clases de C#.

Fuente: investigación propia

4.4. BASE DE DATOS

Ilustración 21 Diagrama Entidad Relación



Fuente: investigación propia

4.4.1. Diccionario de datos

Ilustración 22 Tabla BarriosVeredas

DESKTOP-PUSGKSB\S...bo.BarriosVeredas* ▢ ✕			
	Column Name	Data Type	Allow Nulls
▶🔑	IdBarrioVereda	int	<input type="checkbox"/>
FK	CodMunicipio	int	<input type="checkbox"/>
	Zona	nvarchar(15)	<input type="checkbox"/>
	Distrito	nvarchar(3)	<input type="checkbox"/>
	Descripcion	nvarchar(30)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 23 Tabla Ciudadanos

DESKTOP-PUSGKSB\...- dbo.Ciudadanos ▢ ✕			
	Column Name	Data Type	Allow Nulls
▶🔑	IdCiudadano	int	<input type="checkbox"/>
	Tipoidenticacion	nvarchar(20)	<input type="checkbox"/>
	NumIdentificacion	bigint	<input type="checkbox"/>
	PrimerNombre	nvarchar(20)	<input type="checkbox"/>
	SegundoNombre	nvarchar(20)	<input checked="" type="checkbox"/>
	PrimerApellido	nvarchar(20)	<input type="checkbox"/>
	SegundoApellido	nvarchar(20)	<input checked="" type="checkbox"/>
FK	CodMunNacimiento	int	<input type="checkbox"/>
	FechaNacimiento	datetime	<input type="checkbox"/>
FK	CodBarrioVereda	int	<input type="checkbox"/>
	Celular	bigint	<input type="checkbox"/>
	Email	nvarchar(50)	<input checked="" type="checkbox"/>
	EstadoCivil	nvarchar(20)	<input checked="" type="checkbox"/>
	PuntajeSisben	money	<input type="checkbox"/>
	CabezaFamilia	bit	<input type="checkbox"/>
	Direccion	nvarchar(40)	<input type="checkbox"/>
	Estrato	int	<input type="checkbox"/>
	UsuarioCreador	nvarchar(40)	<input type="checkbox"/>
	FechaCreacion	date	<input type="checkbox"/>
	UsuarioUltimaActualizaci...	nvarchar(40)	<input type="checkbox"/>
	FechaUltimaActualizacion	date	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 24 Tabla Convocatoria

DESKTOP-PUSGKSB\...dbo.Convocatoria			
	Column Name	Data Type	Allow Nulls
	IdConvocatoria	int	<input type="checkbox"/>
FK	CodProyecto	int	<input type="checkbox"/>
	FechaApertura	date	<input type="checkbox"/>
	FechaCierre	date	<input type="checkbox"/>
	Descripcion	nvarchar(200)	<input type="checkbox"/>
	UsuarioCreador	nvarchar(40)	<input type="checkbox"/>
	FechaCreacion	date	<input type="checkbox"/>
	UsuarioUltimaActualizaci...	nvarchar(40)	<input type="checkbox"/>
	FechaUltimaActualizacion	date	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 25 Tabla ConvocatoriaCiudadano

DESKTOP-PUSGKSB\...catoriaCiudadano*			
	Column Name	Data Type	Allow Nulls
	IdConvCiudadano	int	<input type="checkbox"/>
	FechaInscripcion	datetime	<input type="checkbox"/>
FK	CodConvocatoria	int	<input type="checkbox"/>
FK	CodCiudadano	int	<input type="checkbox"/>
FK	CodEstado	int	<input type="checkbox"/>
	UsuarioCreador	nvarchar(40)	<input type="checkbox"/>
	FechaCreacion	date	<input type="checkbox"/>
	UsuarioUltimaActualizaci...	nvarchar(40)	<input type="checkbox"/>
	FechaUltimaActualizacion	date	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 26 Tabla Departamentos

DESKTOP-PUSGKSB\...bo.Departamentos			
	Column Name	Data Type	Allow Nulls
	IdDepartamento	int	<input type="checkbox"/>
	CodDepartamento	int	<input type="checkbox"/>
	DescripcionDep	nvarchar(50)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 27 Tabla Dependencias

DESKTOP-PUSGKSB\...dbo.Dependencias			
	Column Name	Data Type	Allow Nulls
	IdDependencia	int	<input type="checkbox"/>
FK	CodEntidad	int	<input type="checkbox"/>
	DescDependencia	nvarchar(40)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 28 Tabla Entidades

DESKTOP-PUSGKSB\S...1 - dbo.Entidades			
	Column Name	Data Type	Allow Nulls
	IdEntidad	int	<input type="checkbox"/>
FK	CodMunicipio	int	<input type="checkbox"/>
	Nit	bigint	<input type="checkbox"/>
	Descripcion	nvarchar(40)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 29 EstadoCiudadano

Column Name	Data Type	Allow Nulls
 IdEstado	int	<input type="checkbox"/>
DescEstado	nvarchar(20)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 30 Tabla Municipios

Column Name	Data Type	Allow Nulls
 IdMunicipio	int	<input type="checkbox"/>
CodMunicipio	int	<input type="checkbox"/>
DescripcionMun	nvarchar(50)	<input type="checkbox"/>
FK CodDepartamento	int	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 31 Tabla Proyectos

Column Name	Data Type	Allow Nulls
 IdProyecto	int	<input type="checkbox"/>
BpinProyecto	nvarchar(30)	<input type="checkbox"/>
NombreProyecto	nvarchar(200)	<input type="checkbox"/>
ValorProyecto	money	<input type="checkbox"/>
NumDocAprobacion	nvarchar(10)	<input type="checkbox"/>
FechaAprobacion	date	<input type="checkbox"/>
FK CodSector	int	<input type="checkbox"/>
FK CodDependencia	int	<input type="checkbox"/>
NumBeneficiarios	int	<input type="checkbox"/>
Observaciones	nvarchar(200)	<input checked="" type="checkbox"/>
UsuarioCreador	nvarchar(40)	<input type="checkbox"/>
FechaCreacion	date	<input type="checkbox"/>
UsuarioUltimaActualizaci...	nvarchar(40)	<input type="checkbox"/>
FechaUltimaActualizacion	date	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 32 Tabla Requisitos

	Column Name	Data Type	Allow Nulls
	IdRequisito	int	<input type="checkbox"/>
	Descripcion	nvarchar(50)	<input type="checkbox"/>
FK	CodTipo	int	<input type="checkbox"/>
	ValorMinimo	nvarchar(20)	<input checked="" type="checkbox"/>
	ValorMaximo	nvarchar(20)	<input checked="" type="checkbox"/>
	IgualA	nvarchar(20)	<input checked="" type="checkbox"/>
	Si	nvarchar(2)	<input checked="" type="checkbox"/>
	CodConvocatoria	int	<input type="checkbox"/>
	UsuarioCreador	nvarchar(40)	<input type="checkbox"/>
	FechaCreacion	date	<input type="checkbox"/>
	UsuarioUltimaActualizaci...	nvarchar(40)	<input type="checkbox"/>
	FechaUltimaActualizacion	date	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 33 Tabla Sector

	Column Name	Data Type	Allow Nulls
	IdSector	int	<input type="checkbox"/>
	DescSector	nvarchar(20)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 34 Tabla TipoRequisito

	Column Name	Data Type	Allow Nulls
	IdTipo	int	<input type="checkbox"/>
	Descripcion	nvarchar(20)	<input type="checkbox"/>
	CampoBD	nvarchar(20)	<input type="checkbox"/>
	Tabla	nvarchar(20)	<input type="checkbox"/>
	TipoDato	nvarchar(20)	<input type="checkbox"/>
	Lista	nvarchar(20)	<input type="checkbox"/>

Fuente: investigación propia

Ilustración 35 Tabla RequisitosCiudadano

	Column Name	Data Type	Allow Nulls
	IdRequisitoCiudadano	int	<input type="checkbox"/>
FK	CodCovocatoriaCiudada...	int	<input type="checkbox"/>
FK	CodRequisito	int	<input type="checkbox"/>
	RutaDocumento	nvarchar(500)	<input checked="" type="checkbox"/>
	CumpleRequisito	bit	<input type="checkbox"/>
	Validado	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Fuente: investigación propia

5. HERRAMIENTAS UTILIZADAS

5.1. Visual Studio 2019

Es un entorno de desarrollo integrado, creado por la compañía Microsoft y disponible para sistemas operativos Windows, Linux y macOS, y la vez es compatible con múltiples lenguajes de

programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET, fue lanzado en 1997, cuenta con versiones gratis y de venta.

5.2. Microsoft SQL Server

Es un sistema de gestión de base de datos relacional desarrollado como un producto de software con la función principal de almacenar y recuperar datos según lo solicitado por otras aplicaciones

5.3. Postman

Es una herramienta que principalmente nos permite crear peticiones sobre APIs de una forma muy sencilla y poder, de esta manera, probar las APIs. Todo basado en una extensión de Google Chrome. El usuario de **Postman** puede ser un desarrollador que esté comprobando el funcionamiento de una API para desarrollar sobre ella o un operador el cual esté realizando tareas de monitorización sobre un API.

5.4. Azure devops

Planeamiento más inteligente, mejor colaboración y envíos más rápidos con un moderno conjunto de servicios de desarrollo

5.5. React js

Es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en **React**.

6. RESULTADOS

En este capítulo daremos a conocer los resultados obtenidos que abarcan todo el proceso del trabajo realizado.

Los resultados obtenidos se soportan en el presente documento, sobre todo en los objetivos y en cada una de las historias de usuario, las cuales especifican los requerimientos funcionales

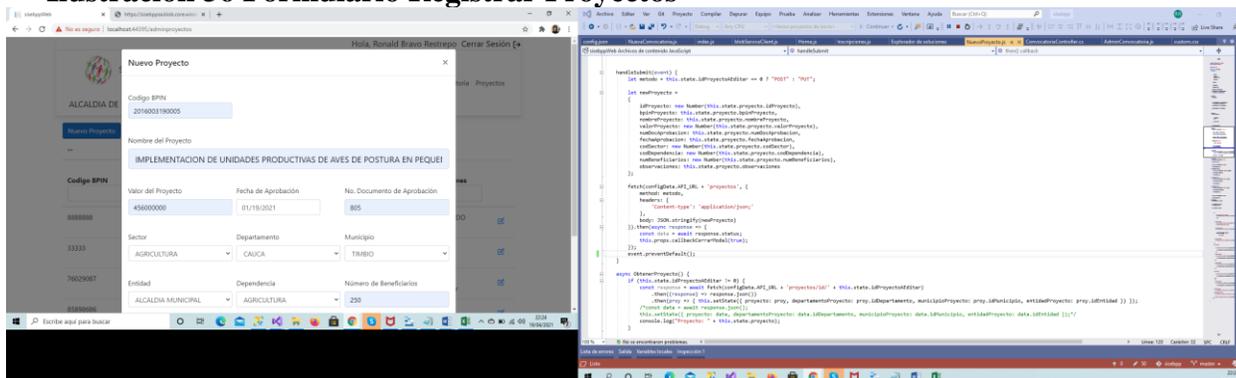
descritos por la alcaldía del municipio de Timbío, Cauca y los requerimientos no funcionales de la aplicación.

6.1 EVIDENCIAS

6.1.1. Formulario registrar proyectos

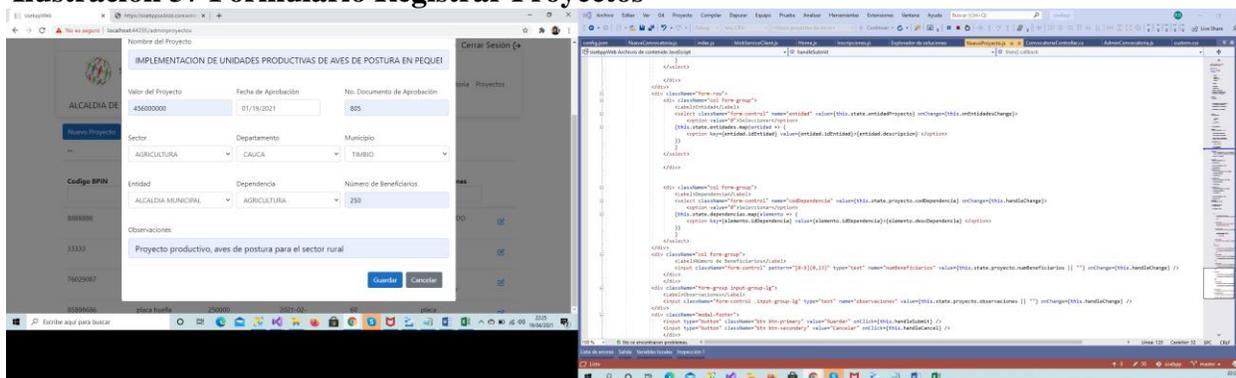
Las siguientes dos ilustraciones muestran el formulario desarrollado para registrar un proyecto desde la plataforma sisebppWeb (Sistema de Selección de Beneficiarios de Proyectos Públicos).

Ilustración 36 Formulario Registrar Proyectos



Fuente: investigación propia

Ilustración 37 Formulario Registrar Proyectos

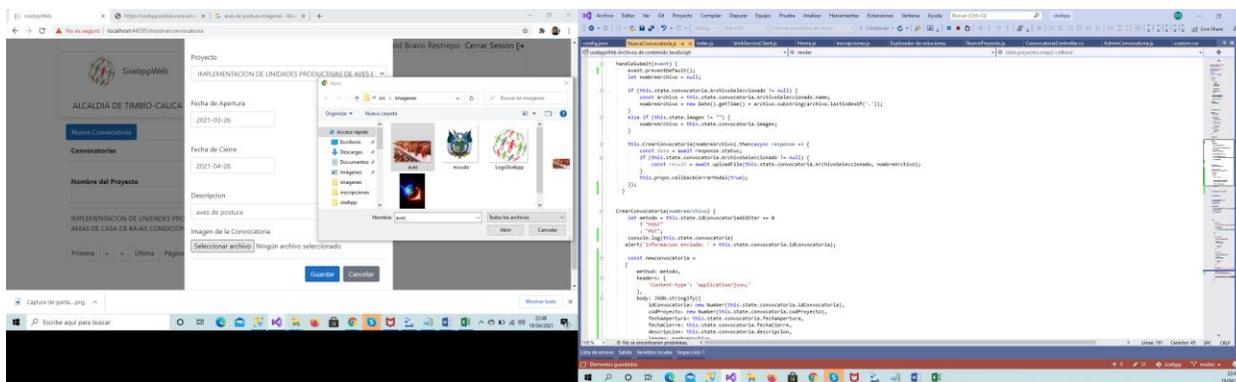


Fuente: investigación propia

6.1.2. Formulario Registrar Convocatoria

La siguiente ilustración muestra el formulario desarrollado para registrar una convocatoria, el formulario permite seleccionar el proyecto a través de una lista, y cargar una imagen relacionada con el mismo.

Ilustración 38 Formulario Registrar Convocatoria

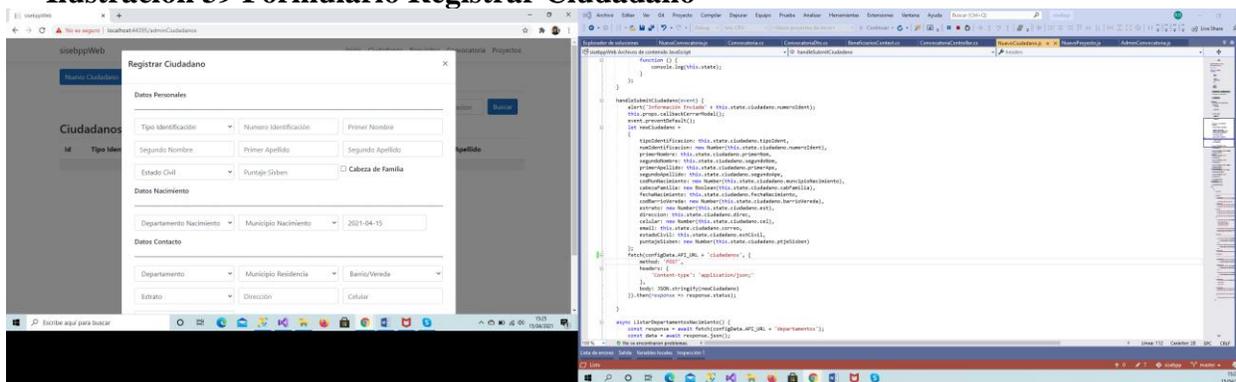


Fuente: investigación propia

6.1.3. Formulario Registrar Ciudadano

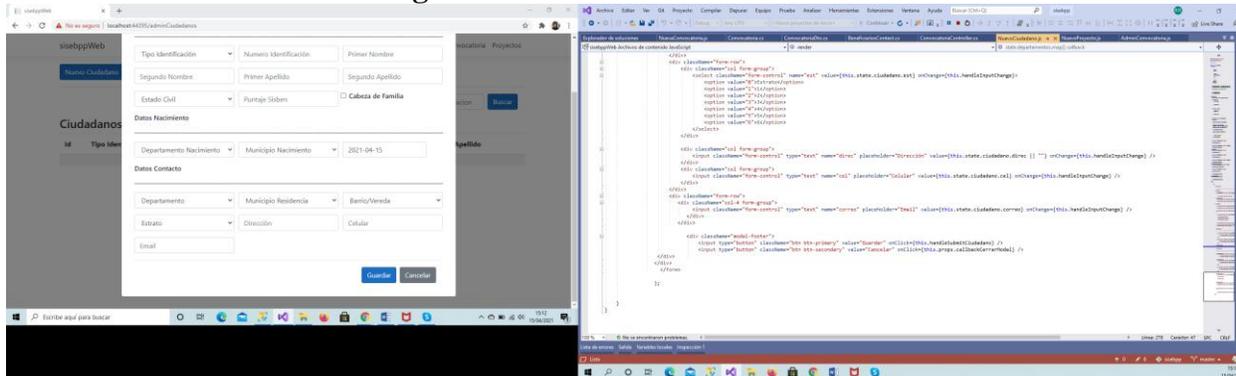
Las siguientes dos ilustraciones muestran el formulario desarrollado para registrar un ciudadano, el formulario permite guardar información relacionada con los datos personales, los datos de nacimiento y los datos de contacto de un ciudadano.

Ilustración 39 Formulario Registrar Ciudadano



Fuente: investigación propia

Ilustración 40 Formulario Registrar Ciudadano

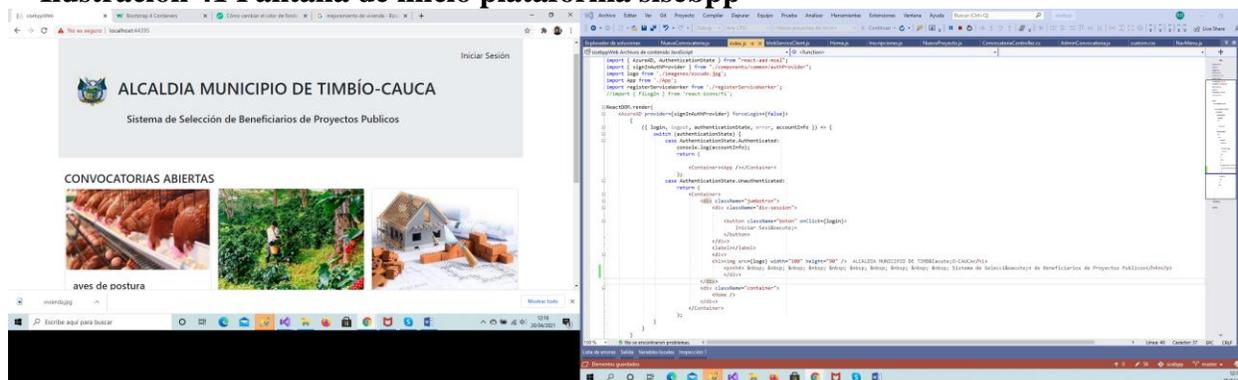


Fuente: investigación propia

6.1.4. Pantalla de inicio plataforma sisebpp

Las siguientes dos ilustraciones muestran la pantalla de inicio de la aplicación Sisebpp (Sistema de Selección de Beneficiarios de Proyectos Públicos). La aplicación muestra las convocatorias abiertas con su respectiva imagen, nombre del proyecto, la fecha de apertura y fecha de cierre., además cuenta con una opción (Iniciar Sesión) que permite a los ciudadanos loguearse y acceder a la aplicación Web.

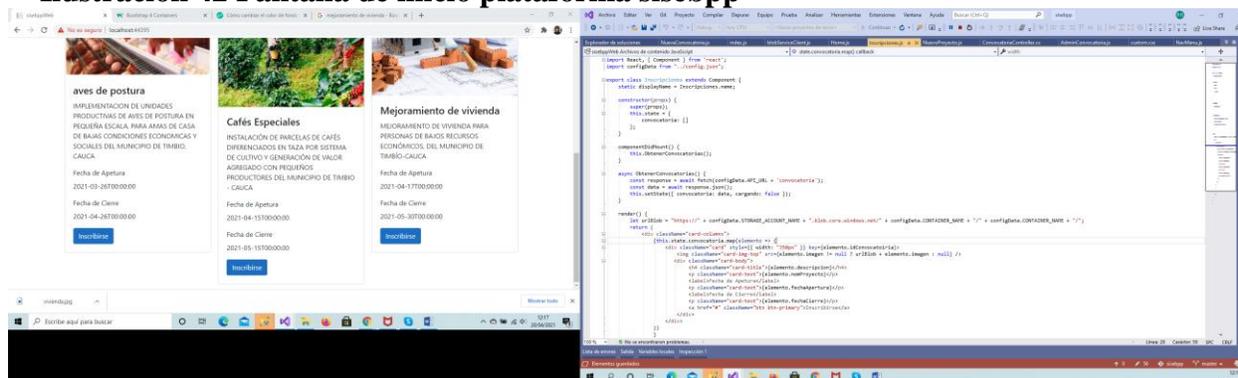
Ilustración 41 Pantalla de inicio plataforma sisebpp



Fuente: investigación propia

Cada Convocatoria cuenta con una opción (Inscribirme) que permita a los ciudadanos inscribirse en una convocatoria desde la pantalla de inicio de la aplicación.

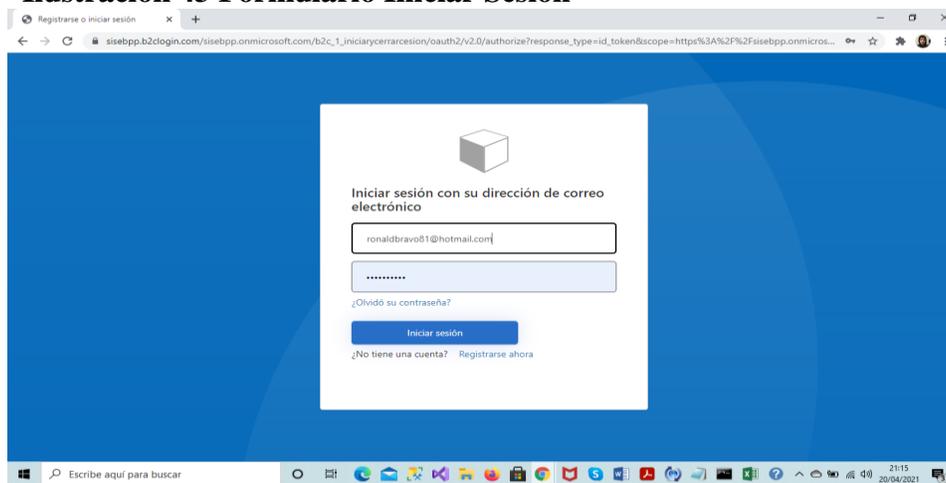
Ilustración 42 Pantalla de inicio plataforma sisebpp



Fuente: investigación propia

6.1.5. Formulario Iniciar Sesión

Ilustración 43 Formulario Iniciar Sesión

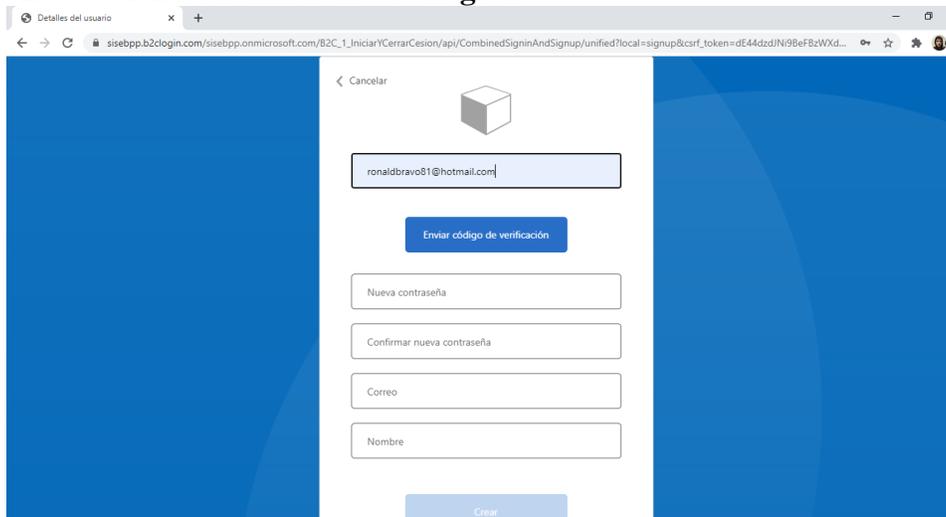


Fuente: investigación propia

6.1.6. Formulario de Registro

Las siguientes ilustraciones muestran el formulario para registrarse y poder ingresar a la plataforma. Este formulario envía un código de verificación al correo electrónico, digitado por el usuario.

Ilustración 44 Formulario de Registro



Fuente: investigación propia

El usuario debe digitar en el formulario de registro el código enviado al correo electrónico para que este lo pueda validar, ingresar los datos restantes en el formulario y finalmente registrarse e ingresar a la plataforma Web.

Ilustración 45 Formulario de Registro

Se ha enviado el código de verificación a su Bandeja de entrada. Cópielo en el siguiente cuadro de entrada.

ronaldravo01@hotmail.com

462865

Verificar código Enviar código nuevo

Nueva contraseña

Confirmar nueva contraseña

Correo

Nombre

Crear

Fuente: investigación propia

Ilustración 46 formulario de Registro

Dirección de correo electrónico comprobada. Puede continuar.

ronaldravo01@hotmail.com

Cambiar correo electrónico

.....

.....

ronaldravo01@hotmail.com

Ronald Bravo Restrepo

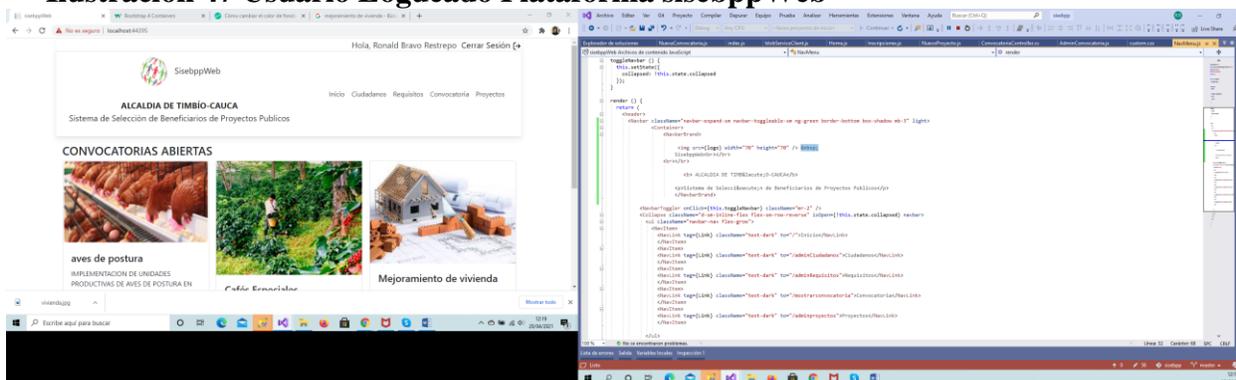
Crear

Fuente: investigación propia

6.1.7. Usuario Logueado Plataforma sisebppWeb

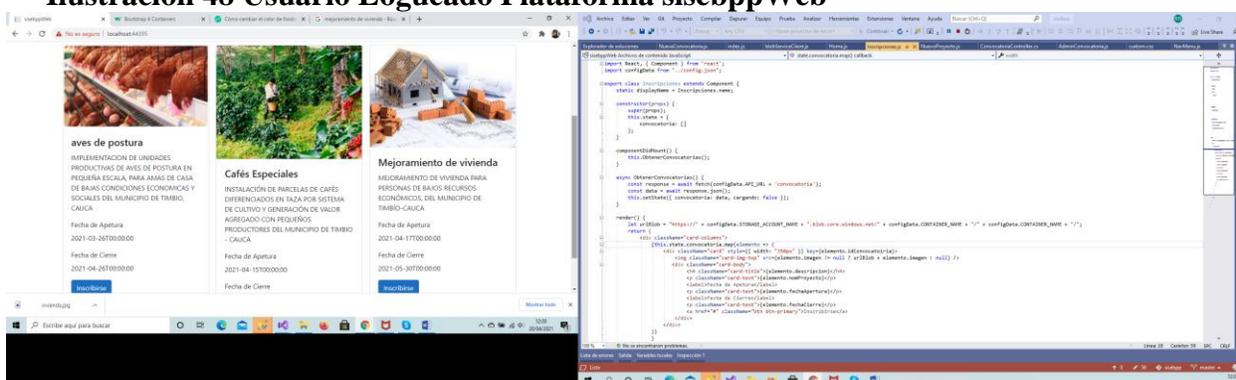
Las siguientes dos ilustraciones muestran el usuario logueado en la plataforma, la cual permite al usuario administrador, administrar la información de los proyectos, de las convocatorias, de los ciudadanos y de los beneficiarios de los proyectos.

Ilustración 47 Usuario Logueado Plataforma sisebppWeb



Fuente: investigación propia

Ilustración 48 Usuario Logueado Plataforma sisebppWeb



Fuente: investigación propia

7. CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

- Teniendo en cuenta el desarrollo del trabajo y los requerimientos exigidos por la alcaldía del municipio de Timbío, Cauca, especificados en cada una de las historias de usuario, se puede determinar que se cumplió con el objetivo propuesto.
- La aplicación desarrollada es una herramienta de gran beneficio para la Alcaldía del municipio de Timbío, Cauca, dado que permite apoyar a las diferentes dependencias de la institución, en el proceso de selección de los beneficiarios de los proyectos públicos y de esta manera realizar una mejor distribución de los recursos entre la población, de una forma más equitativa.
- Esta aplicación es muy importante para la población del municipio de Timbío, Cauca, ya que les permite a las personas, sobre todo a aquellas que viven lejos de la zona urbana, registrarse desde la plataforma, acceder a las convocatorias, poder inscribirse y cargar los documentos exigidos, desde el lugar donde se encuentren o residen.
- El desarrollo de este modelo de aplicación es de gran ayuda para los funcionarios, ya que les facilita el trabajo y les permite registrar las convocatorias y configurar sus requisitos, registrar los ciudadanos, registrar los proyectos e inscribir a los ciudadanos en las convocatorias, y tener toda esta información almacenada en una base de datos, además de la información de las personas beneficiadas en otros proyectos, ejecutados por las diferentes dependencias de la Alcaldía del municipio de Timbío, Cauca.
- El desarrollo de esta herramienta Web constituye una solución tecnológica para la Alcaldía del municipio de Timbío, Cauca, ya que permite cruzar la información de las personas inscritas en las nuevas convocatorias, con la información de los beneficiarios de otros proyectos y generar un listado de potenciales beneficiarios, teniendo en cuenta los requisitos exigidos para cada convocatoria y las personas beneficiadas en otros

proyectos, esto con el fin de apoyar en la toma de decisiones al momento de escoger los beneficiarios.

- Las aplicaciones web son de gran importancia en el sector público, dado que establecen un canal de comunicación directo con la comunidad y ayudan en la transparencia de los procesos.

7.2. RECOMENDACIONES

- Se sugiere a la Alcaldía del municipio de Timbío, Cauca, promover entre la comunidad el uso de la plataforma para inscribirse en las convocatorias y participar en la selección de los beneficiarios de los proyectos.
- Capacitar a los funcionarios de las diferentes dependencias para darle buen uso a la aplicación.
- Planear una fase de información histórica de los beneficiarios de proyectos para ingresarlos en la aplicación, esto con el fin de mejorar la funcionalidad de selección de los beneficiarios.

8. BIBLIOGRAFÍA

1. Amaro Calderón, S. D., & Valverde Rebaza, J. C. (2007). *Metodologías Ágiles*. Universidad Nacional de Trujillo.
2. Becerril Isidro, J., Vallejo Lassard, A. P., Lumbreras Sotomayor, A., Chávez Ojeda, G. A., & Duk Sánchez, A. R. (2012). *La Web 2.0*. Simón Bolívar.
3. Beck, K., & Andres, C. (2004). *Extreme Programming Explained*. Addison-Wesley Professional.
4. Colombia. Departamento Administrativo para la Prosperidad Social. (2019). *Manual Operativo Familias en Acción*. Bogotá: Diario Oficial .
5. Colombia. Departamento Nacional de Planeación. (2016). *Declaración de importancia Estratégica del Sistema de Identificación de Potenciales Beneficiarios (SISBEN IV)*. Colombia: Diario Oficial.
6. Colombia. Departamento Nacional de Planeación. (2018). *Manual de Funcional del Sistema Unificado de Inversiones y finanzas publicas*. Bogotá: Diario Oficial .
7. Colombia. Ministerio de Salud y Protección Social. (2015). *Decreto 2353, Por el cual se unifican y actualizan las reglas de afiliación al Sistema General de Seguridad Social en Salud*. Bogotá: Diario Oficial.
8. Colombia. Unidad para las Víctimas . (2018). *MANUEL DE INGRESO AL REGISTRO UNICO DE VICTIMAS*. Bogotá: Diario Oficial .
9. Desarrolloweb. (2020). *Qué es MVC*. Obtenido de Desarrolloweb: <https://desarrolloweb.com/articulos/que-es-mvc.html>
10. LAÍNES Fuentes, J. R. (2015). *Desarrollo de Software Ágil Extreme Programming y Scrum*. IT Campus Academi.
11. MELÉNDEZ VALLADAREZ, S. M., GAITAN, M. E., & PÉREZ REYES, N. N. (2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION*. UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA.
12. Proaño Castro, M. F., Orellana Contreras, S. J., & Martillo Pazmillo , I. O. (2018). Los sistemas de información y su importancia en la Transformación Digital de la empresa actual. *Espacios*, 39 (45), 4. Obtenido de <https://www.revistaespacios.com/a18v39n45/a18v39n45p03.pdf>
13. Ricardo, C. M. (2009). *Bases de Datos*. MacGraw-Hill Companies.

14. Sanchez, w. (2011). *La Usabilidad en Ingeniería de Software*. Universidad Don Bosco.
15. Wikipedia. (2021). *Portabilidad*. Obtenido de Wikipedia:
<https://es.wikipedia.org/wiki/Portabilidad>