



**UNIVERSIDAD ESTATAL DE MILAGRO
FACULTAD DE CIENCIAS E INGENIERÍA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE GRADO EN LA CARRERA DE INGENIERO EN
SISTEMAS COMPUTACIONALES**

PROYECTO TÉCNICO

**TEMA: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA
APLICACIÓN MÓVIL Y UN SISTEMA WEB PARA GESTIONAR EL
INGRESO DE VISITAS A LA URBANIZACIÓN “LA GEMA”**

Autores:

Sr. Cuadrado Hidalgo Jefferson Alejandro

Sr. Zambrano Domínguez Israel Moisés

Tutor: Mgtr. López Bermúdez Ricauter Moisés

Milagro, Junio 2022

ECUADOR

DEDICATORIA

El presente trabajo se lo dedico principalmente a Dios, por ser mi ayudador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres Guido y Yina quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer las adversidades porque Dios está conmigo siempre.

A mis hermanos Stalin, David y Damaris por su cariño y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento gracias. A toda mi familia porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

Finalmente, una dedicatoria muy especial a mi esposa por su apoyo incondicional durante este último año de estudio, ya que sin su ayuda y amor no lo hubiera logrado.

Israel Zambrano Domínguez

Dedico este proyecto a Dios, porque ha estado conmigo en cada paso que doy llenándome de salud, sabiduría y fortaleza en todo momento. A mis familiares, ya que ellos son un pilar fundamental en mi vida, son una fuente de inspiración y quienes me motivan a mejorar día tras día como ser humano y futuro profesional.

Un especial sentimiento de gratitud hacia mis amados padres, cuyas palabras de aliento retumban en mis oídos y me ayudan a seguir en la lucha para llegar a la meta. También dedico esta disertación a mis amigos, compañeros, docentes de la universidad y a cada una de las personas que han estado a mi lado durante mi proceso de formación académica.

Jefferson Cuadrado Hidalgo

AGRADECIMIENTO

Agradecemos a Dios por bendecirnos la vida, por guiarnos a lo largo de nuestra existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad.

Gracias a nuestros padres, por ser los principales promotores de nuestros sueños, por confiar y creer en nuestras expectativas, por los consejos, valores y principios que nos han inculcado.

Agradecemos a nuestros docentes de la Universidad Estatal de Milagro, por haber compartido sus conocimientos a lo largo de la preparación de nuestra profesión, de manera especial a nuestro tutor el Mgtr. López Bermúdez Ricauter Moisés quien nos ha guiado con su paciencia, y su rectitud como docente durante todo este proceso, y a los directivos de residentes de la Urbanización “La Gema” por su valioso aporte para nuestro proyecto de investigación.

ÍNDICE GENERAL

DEDICATORIA	2
AGRADECIMIENTO	3
ÍNDICE GENERAL	4
ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	7
RESUMEN	1
ABSTRACT	2
CAPÍTULO 1	3
1. INTRODUCCIÓN	3
1.1. Planteamiento del problema	4
1.2. Objetivos	5
1.2.1. Objetivo General	5
1.2.2. Objetivos Específicos	5
1.3. Alcance	5
1.4. Estado del arte	5
1.4.1 Artículos y tesis que aportan a mi proyecto	5
1.4.2 Conceptos para el desarrollo de la aplicación web	9
1.4.3 Conceptos para el desarrollo de la aplicación móvil	10
1.4.4 Tecnologías utilizadas	15
Lenguajes de programación	15
Frameworks	17
Base de Datos	22
CAPÍTULO 2	24
2. METODOLOGÍA	25
2.1 Desarrollo de Software con Metodología en Cascada	25
2.1.1. Fases del Modelo de Cascada	26
Fase de Análisis de Requerimientos	26
Fase de Diseño	26
Fase de Codificación	27
Fase de Pruebas	27
	4

Fase de Mantenimiento	27
CAPÍTULO 3	28
3. PROPUESTA DE SOLUCIÓN	28
3.1 Tema	28
3.2 Descripción de la propuesta	28
3.3. Desarrollo de Software con Metodología en Cascada	28
3.1.1. Fases de la Metodología en Cascada	28
Fase de Análisis de los requerimientos	28
Fase de Diseño	29
Fase de Codificación	60
Fase de Pruebas	65
Fase de Mantenimiento	65
CONCLUSIONES	66
RECOMENDACIONES	67
REFERENCIAS BIBLIOGRÁFICAS	68
ANEXOS	70

ÍNDICE DE FIGURAS

<i>Fig. 1 Descripción gráfica del sistema de rejillas de Bootstrap</i>	21
<i>Fig. 2 Modelo de Cascada</i>	26
<i>Fig. 3 Diagrama de casos de uso del administrador</i>	30
<i>Fig. 4 Diagrama de casos de uso del administrador</i>	31
<i>Fig. 5 Modelo entidad/relación del sistema.</i>	32
<i>Fig. 6 Pantalla de Login</i>	46
<i>Fig. 7 Menú de la aplicación</i>	47
<i>Fig. 8 Listado de alícuotas</i>	47
<i>Fig. 9 Pantalla para mantenimiento de residentes</i>	48
<i>Fig. 10 Pantalla para listado de tarjetas</i>	49
<i>Fig. 11 Formulario para agregar una nueva tarjeta.</i>	49
<i>Fig. 12 Listado de urbanizaciones.</i>	50
<i>Fig. 13 Inicio de la App</i>	51
<i>Fig. 14 Login de acceso a la aplicación</i>	52
<i>Fig. 15 Sección de Noticias</i>	53
<i>Fig. 16 Pantalla de Inicio</i>	54
<i>Fig. 17 Listado de Invitaciones</i>	55
<i>Fig. 18 Compartir Invitación</i>	56
<i>Fig. 19 Crear Invitación</i>	57
<i>Fig. 20 Contactos Guardados</i>	58
<i>Fig. 21 Pantalla de Mensajes</i>	59
<i>Fig. 22 Pantalla de Perfil</i>	60

ÍNDICE DE TABLAS

<i>Tabla 1. Operaciones RESTful</i>	10
<i>Tabla 2. Principales entornos de desarrollo según el sistema operativo</i>	13
<i>Tabla 3. Comparativa del tiempo que tarda en renderizar Vue.js frente a React</i>	19
<i>Tabla 4. Descripción de la tabla de Género</i>	33
<i>Tabla 5. Descripción de la tabla Usuario</i>	34
<i>Tabla 6. Descripción de la tabla roles de usuario</i>	34
<i>Tabla 7. Descripción de la tabla rol</i>	35
<i>Tabla 8. Descripción de la tabla Log de usuarios</i>	35
<i>Tabla 9. Descripción de la tabla persona</i>	36
<i>Tabla 10. Descripción de la tabla residente</i>	37
<i>Tabla 11. Descripción de la tabla familiar</i>	38
<i>Tabla 12. Descripción de la tabla Distributivo de persona</i>	39
<i>Tabla 13. Descripción de la tabla vivienda</i>	40
<i>Tabla 14. Descripción de la tabla Urbanización</i>	40
<i>Tabla 15. Descripción de la tabla Modelo de vivienda</i>	41
<i>Tabla 16. Descripción de la tabla Alícuota</i>	42
<i>Tabla 17. Descripción de la tabla Post de información</i>	43
<i>Tabla 18. Descripción de la tabla Mensaje de la App</i>	43
<i>Tabla 19. Descripción de la tabla invitación de la App</i>	44
<i>Tabla 20. Descripción de la tabla Estado de la invitación de la App</i>	44
<i>Tabla 21 Descripción de la tabla etapa de la aplicación web</i>	45

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL Y UN SISTEMA WEB PARA GESTIONAR EL INGRESO DE VISITAS A LA URBANIZACIÓN “LA GEMA”

RESUMEN

El presente proyecto de titulación, se basa en el análisis, diseño e implementación de una aplicación móvil y un sistema Web que permita gestionar el ingreso de visitantes a la urbanización “La Gema” de la ciudad de Guayaquil, con el objetivo de reducir la inseguridad monitoreando el acceso de las personas que ingresan a la urbanización, tanto familiares de los residentes como de personas externas. Para desarrollar este proyecto se utilizó la metodología de cascada y el modelado de datos entidad-relación que ayudó al análisis y diseño estandarizado en base a los requerimientos del usuario. En las urbanizaciones privadas es imprescindible tener mayor control sobre las visitas, para la seguridad de los residentes, es por eso que se llevó a cabo el desarrollo de una aplicación móvil para que los visitantes puedan acceder fácilmente con un código de validación QR y un sistema Web para la administración de la urbanización donde podrán gestionar y crear los usuarios de los residentes para que puedan acceder mediante la aplicación. De manera que, los residentes se sentirán con mayor tranquilidad sabiendo que existe un control adecuado sobre las personas que ingresan a la urbanización, y además evitan riesgos delictivos y aglomeración de visitantes.

PALABRAS CLAVE: Aplicación móvil, Sistema Web, Urbanización, Control de visitas,

ANALYSIS, DESIGN AND IMPLEMENTATION OF A MOBILE APPLICATION AND A WEB SYSTEM TO MANAGE THE ENTRY OF VISITORS TO THE URBANIZATION "LA GEMA"

ABSTRACT

This degree project is based on the analysis, design and implementation of a mobile application and a Web system that allows managing the entry of visitors to the urbanization "La Gema" in the city of Guayaquil, with the aim of reducing insecurity. monitoring the access of the people who enter the urbanization, both relatives of the residents and external people. To develop this project, the waterfall methodology and entity-relationship data modeling were used, which helped the analysis and standardized design based on user requirements. In private urbanizations it is essential to have greater control over visits, for the safety of residents, that is why a mobile application was developed so that visitors can easily access with a QR validation code and a system Web for the administration of the urbanization where they can manage and create the users of the residents so that they can access through the application. In this way, residents will feel more at ease knowing that there is adequate control over the people who enter the urbanization, and they also avoid criminal risks and agglomeration of visitors.

KEY WORDS: Mobile application, Web System, Urbanization, Control of visits,

CAPÍTULO 1

1. INTRODUCCIÓN

Actualmente, existen muchas urbanizaciones en el país que brindan servicios de vivienda, que están asociados a servicios complementarios para garantizar un buen vivir. Las funciones administrativas, de control y de seguridad trabajan como unidad de negocio, al garantizar una gestión óptima y puntual, se convierten en un servicio indispensable para los habitantes. Las aplicaciones móviles hoy en día brindan acceso inmediato a los usuarios, conectándolos a la información al instante e incluso en línea. Con las aplicaciones móviles, se reduce el tiempo y las largas colas en la gestión de servicios, pagos, eventos y más.

En este caso la urbanización “La Gema”, no cuenta con un sistema que les permita gestionar y controlar el acceso de los visitantes, sino que mantiene una gestión manual y tradicional. La falta de un sistema automatizado ocasiona la inconformidad de los residentes hacia los administradores debido a la pérdida de tiempo, aglomeración de visitantes, pérdida de documentos de identidad y desorden en la garita.

Por lo anterior, el presente proyecto de titulación tiene como objetivo desarrollar un sistema Web con acceso a una aplicación móvil que permita gestionar el ingreso de visitas con un código de validación QR (Quick Response) para mejorar el tiempo de ingreso a la Urbanización “La Gema”. También facilitará informes y reportes de control de accesos de visitas a la urbanización de acuerdo a las necesidades del administrador.

1.1. Planteamiento del problema

En la actualidad el uso de aplicaciones móviles es cada vez más indispensable, debido a que se encuentran disponibles en todo momento a través de los Smartphone, permitiendo la facilidad de realizar transacciones de manera rápida y efectiva.

La seguridad y el bienestar de los residentes es una de las prioridades de las urbanizaciones, debido a los múltiples riesgos delictivos que enfrentan. Por ello, resulta imprescindible la figura del vigilante de seguridad en las urbanizaciones, ya que gracias a él se reduce la vulnerabilidad de las urbanizaciones frente a hurtos, robos o accesos indebidos, con el objeto de aprovechar y deteriorar los servicios de la misma, como pueden ser el mobiliario urbano, piscinas, pistas de deporte, etc.

Unas de las principales preocupaciones de las urbanizaciones en el Ecuador son los accesos indeseados, ya que en muchos casos debido a la aglomeración de visitantes que esperan en garita, el guardia no realiza el debido control de entrada o a su vez se demora en registrar y validar el documento de identidad del visitante, causando inconformidad a los residentes.

Este es el caso de la urbanización “La Gema” quien no cuenta con un sistema adecuado que les permita gestionar y automatizar el control de visitas, por lo que muchos residentes están inconformes con la administración debido a que el guardia muchas veces no se abastece ya que es el encargado de realizar todo el control de entrada en garita, ocasionando pérdida de tiempo, aglomeración de visitantes, pérdida de documentos de identidad y desorden en garita.

En relación a lo expuesto, se llevó a cabo una entrevista con el Ing. Roberto León actual presidente de los residentes de la urbanización “La Gema”, quien, a su vez en una reunión mantenida con los demás habitantes, se pudo determinar que la principal causa que está generando el problema es que se realiza de forma manual el proceso de control de visitas y eso se torna lento y tedioso, causando enojo en los residentes que esperan a sus visitantes.

Debido a la problemática expuesta anteriormente, se propone una aplicación móvil de control de acceso de visitas, la cual a través de ella permitirá crear invitaciones con fecha y hora establecida a través de un código QR (Quick Response) de acceso único que podrá ser compartido a familiares y amigos. Además, se creará un sistema web para la

administración de la urbanización donde podrán gestionar y crear los usuarios de los residentes para el acceso a la App y poder sacar los diferentes reportes de control.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un sistema Web con acceso a una aplicación móvil que permita gestionar el ingreso de visitas con un código de validación QR (Quick Response) para mejorar el tiempo de ingreso a la Urbanización “La Gema”.

1.2.2. Objetivos Específicos

- Definir los requerimientos y necesidades de la urbanización de estudio para conocer la situación actual de los procesos de ingreso de visitas e identificar los módulos que contendrá el sistema.
- Aplicar la metodología de desarrollo adecuada que permita implementar las mejores prácticas en el desarrollo del sistema Web y la aplicación móvil.
- Proporcionar dentro de la aplicación móvil una opción que facilite enviar invitaciones de visitas a través de un código de acceso QR que permita validar el ingreso a la Urbanización “La Gema”.

1.3. Alcance

El presente proyecto tiene como objetivo satisfacer las necesidades de los usuarios y administradores de la etapa La Gema de la urbanización la joya mediante la creación de la aplicación web mejorando su administración en torno al manejo de la información del pago de las alcuotas, registro de usuarios e información de actividades a realizar. Así como también la creación de una aplicación móvil para mejorar la seguridad de acceso de visitas en la Urbanización. Para lo cual se consideró la creación de diferentes módulos que facilitan el diseño y manejo de la aplicación.

1.4. Estado del arte

1.4.1 Artículos y tesis que aportan a mi proyecto

1. “Diseño de un sistema de administración en urbanizaciones de la ciudad de Milagro”

El objetivo principal de este proyecto fue la optimización en el procesamiento de la información que se genera diariamente en la urbanización “Quinta Patricia” del cantón

Milagro, ya que, como se describe en el documento, los registros del pago de alcúotas se llevan parcialmente de forma manual ayudados únicamente con hojas de Microsoft Excel almacenadas en un ordenador de uso básico (Chimbo, 2019).

En cuanto a la justificación de su proyecto, el autor señala que con base en encuestas y observación directa llegó a la conclusión de que se necesita implementar un software de gestión administrativa en primera instancia, para agilizar la consulta de datos ya que si bien es cierto el pago de alcúotas es una transacción sumamente básica, se necesita llevar un registro de cada uno de los pagos por persona lo cual generaría un gran volumen de datos que con la ayuda de herramientas de ofimática como Excel se volvería una tarea sumamente compleja. Además, para que los habitantes puedan consultar en cualquier momento el registro de cada uno de sus pagos y la forma en la que se están gestionando sus aportaciones ya que toda esta información de carácter público se encontraría almacenada en la base de datos.

La solución que se pretende dar con este proyecto además aporta al desarrollo sostenible de la urbanización ya que se eliminaría el desperdicio de hojas de papel y la contaminación por residuos eléctricos y electrónicos que en este caso correspondían a impresoras, tinta, tóner, etc. En ese sentido podemos concluir que el proyecto es factible por tres razones, en primer lugar, como ya lo mencionamos es eco amigable, en segundo lugar, es un proyecto no tan costoso ya que para su desarrollo se pretende utilizar herramientas de código abierto como PHP y MySQL por lo cual la tasa de retorno anual es del 42,06% y, por último, el impacto social es positivo pues con base en las encuestas el nivel de satisfacción de los usuarios con respecto al tratamiento de sus datos es del 70%. (Chimbo, 2019).

La presente revisión de la literatura además aporta al desarrollo de nuestro proyecto ya que define las bases bibliográficas para poder indagar en cuanto a cómo debemos desarrollar el módulo de administración. Además, provee una sólida explicación en cuanto a la metodología de desarrollo en cascada que nos sirve de referencia para tener en cuenta a la hora de elegir la metodología de desarrollo que vamos a implementar en nuestra propuesta de titulación.

2. “Desarrollo e Implementación de una aplicación para la administración de la Urbanización La Joya, etapa Murano y control de alcúotas”

El objetivo principal de este ambicioso proyecto fue el de dar soporte al área de administración desarrollando una aplicación cliente-servidor para gestionar la información del pago de alcuotas, registro de villas, reservación de espacios recreativos y el registro de usuarios. Así mismo, diseñar una aplicación móvil nativa que ayude a mejorar la seguridad en la urbanización mediante un sistema de alertas (Correa & Castro, 2016).

El sistema de alertas que se desarrolló en la aplicación móvil consta de un botón de pánico para poder recibir ayuda inmediata del guardia de la urbanización frente a cualquier conflicto que se pueda suscitar. En ese sentido se prevé que la seguridad de la urbanización mejoraría pues cada habitante tendría un medio a través del cual comunicarse frente a una potencial amenaza y que, además como un filtro adicional, el autor del proyecto pretende gestionar la información de cada uno de los visitantes a través de una base de datos para que el guardia que se encuentra en la entrada no tenga que buscar entre largos registros impresos en papel sino únicamente digitar el número de cédula del visitante y si está registrado previamente se le permitirá el acceso.

En cuanto a la justificación del proyecto (Correa & Castro, 2016) señalan que desarrollaron este sistema con la finalidad de mejorar el proceso de gestión administrativa de la urbanización, ya que si bien es cierto, la forma en que lo hacían hasta el momento de realizar el análisis de requerimientos no es del todo obsoleta, es una forma poco convencional considerando el hecho de que hoy en día tenemos la posibilidad de automatizar estos procesos con la ayuda de herramientas tecnológicas de código abierto que ya en la práctica no representan una inversión cuantiosa ni difícil de mantener. Así mismo, el sistema en el módulo de seguridad pretende dar soporte al control de visitantes para evitar el ingreso de personas no deseadas (Correa & Castro, 2016).

Este proyecto además aporta de manera significativa al desarrollo de nuestra propuesta de titulación ya que aparte del módulo de administración también contiene una pequeña parte de lo que se pretende implementar en nuestra propuesta pues como se describe en el documento, ellos implementaron un “botón de pánico” que es una capa adicional al sistema de seguridad que posee la urbanización. Esta iniciativa nos sirve como punto de referencia para la realización de un botón de pánico mejorado,

tomando en consideración las situaciones que no fueron contempladas en la versión inicial.

3. “Implementación de un dispositivo de acceso de control inteligente de gestión de visitantes de una urbanización privada mediante una aplicación móvil usando recursos no renovables”

Este proyecto tiene como objetivo la automatización del control de acceso a la urbanización Toledo perteneciente a “Villa España 2” en la ciudad de Guayaquil, la solución a implementar se da mediante la instalación de sensores en la entrada del residencial incrustados en una placa Arduino que receptorá la información de los visitantes y la almacenará en una base de datos para posteriormente enviarla al dispositivo móvil del residente que recibirá la visita y será este quien determine si la persona puede ingresar o no al residencial (Laborde & Paredes, 2020).

En cuanto al alcance del proyecto, los autores pretenden implementar una solución a los inconvenientes que se generan debido a la falta de adaptación al uso de herramientas tecnológicas durante el proceso de verificación en el ingreso de personas en general a la urbanización. La solución propuesta, se divide en dos componentes, el hardware cuyo ensamblaje incluye un microcontrolador que será el procesador del dispositivo y un panel fotovoltaico para alimentar el sistema de energía eléctrica. Y el software, que es una aplicación móvil nativa utilizada como puerta de acceso a la base de datos que contiene información sobre cada uno de los habitantes de la urbanización.

Uno de los factores de mayor importancia en cuanto a la justificación del proyecto es la inseguridad que se está evidenciando a nivel nacional, motivo por el cual los administradores de las urbanizaciones están optando por buscar soluciones tecnológicas que sirvan como un filtro adicional en sus sistemas de control de acceso al inmueble y de esta forma asegurar que cada uno de los habitantes pueda sentirse confiado dentro y fuera de su vivienda en materia de seguridad ciudadana. Adicionalmente a esto, la solución propuesta por los autores contribuye al desarrollo sostenible de la urbanización ya que el dispositivo electrónico utiliza energía eco amigable y no genera gastos adicionales mayores luego de su implementación (Laborde & Paredes, 2020).

Con base en el análisis de este documento es posible determinar la importancia de un adecuado control de acceso a la urbanización, ya que si existieran falencias en este

proceso no se podría asegurar la integridad física de los habitantes ni de sus bienes materiales, por lo cual, el presente documento aporta información detallada de cómo debería funcionar un sistema para el control de acceso a una urbanización y con base en esto podemos los autores realizar ciertas mejoras al prototipo ya establecido.

1.4.2 Conceptos para el desarrollo de la aplicación web

Aplicación Web

En la actualidad se ha vuelto imprescindible el que un establecimiento ya sea público o privado cuente con un servicio de internet fijo debido a la necesidad de acceder a información en la web, en ese sentido las aplicaciones que funcionan sobre esta plataforma están revolucionando la manera en la que se almacenan y procesan los datos.

Un sistema web o también llamado aplicación web, es una aplicación de software que puede ser usada accediendo a un servidor web remoto por medio de internet o un servidor web local usando una intranet con ciertas ventajas significativas frente a las aplicaciones de escritorio como, por ejemplo: la posibilidad de acceder a la aplicación desde cualquier parte del mundo y que varios usuarios pueden ingresar a la misma aplicación de forma simultánea. (Paredes Colmenar & Millanes Santos, 2020)

En cuanto a su interfaz gráfica, es muy similar a la de un sitio web y en algunos casos se utilizan métricas de UX para que de esta forma el lapso de aprendizaje del usuario sobre el manejo de la aplicación sea el más corto posible. Una aplicación web es relativamente más fácil de aprender que una aplicación de escritorio por el simple hecho de que los usuarios están constantemente accediendo a sitios web y conocen la estructura de estas páginas, por lo tanto, el uso de una aplicación web se vuelve intuitivo.

API RESTful

Mediante llamadas a la API de RESTful vamos a integrar la interfaz gráfica de usuario de la aplicación móvil con toda la lógica de programación situada en un servidor web que a su vez gestiona el acceso a la base de datos que contiene información sobre la urbanización en general, de esta forma logramos tener una independencia de tecnologías, separamos cliente/servidor, conseguimos que el sistema cumpla con los criterios de escalabilidad y sea fácil de mantener. (Ahmad, 2017)

Una API de RESTful, es una interfaz de programación de aplicaciones que se adapta a la arquitectura REST y permite la interoperabilidad con los servicios en ambiente web de RESTful.

<i>Operación</i>	<i>Método HTTP</i>	<i>URL</i>	<i>Parámetros</i>	<i>Resultado</i>
Listar	GET	/{recurso}	No aplica	Lista del tipo de recurso
Crear	POST	/{recurso}	Dentro del cuerpo en el POST	Se crea un nuevo recurso
Leer	GET	/{recurso}/{recurso_id}	No aplica	Devuelve un recurso con un id específico
Actualizar	PATCH /PUT	/{recurso}/{recurso_id}	Se pasan en la URL	Se actualiza /reemplaza el recurso según su id
Borrar	DELETE	/{recurso}/{recurso_id}	No aplica	Se elimina el recurso por id

Tabla 1. Operaciones RESTful
Fuente: (Ahmad, 2017)

Las API son un conjunto de protocolos y definiciones que se utilizan para diseñar e integrar el software de las aplicaciones. Esta interacción sin embargo no se da de humano a máquina y viceversa como es habitual, sino que en este caso la información viaja de máquina a máquina. Comúnmente es considerada como el contrato entre el usuario y el proveedor de información, donde el usuario establece los parámetros y el tipo de método HTTP a utilizar esperando una respuesta que puede ser un código de estado (en el caso de una llamada a crear, editar, eliminar un recurso), un recurso en específico o el listado de recursos almacenados en la base de datos que gestiona la API.

Mediante llamadas a la API de RESTful se pretende integrar la interfaz gráfica de usuario de la aplicación móvil con toda la lógica de programación situada en un servidor remoto que a su vez gestiona una base de datos con información relevante sobre la urbanización en general.

1.4.3 Conceptos para el desarrollo de la aplicación móvil

Aplicación móvil

Decidimos diseñar una aplicación móvil como puerta de enlace a los servicios web de la urbanización “La Joya” etapa Murano debido a que estos sistemas son de fácil implementación y porque además al contar únicamente con las opciones principales se espera que la curva de aprendizaje sobre su uso sea mucho más baja en comparación con una aplicación web que integre todas las funcionalidades del sistema.

Una aplicación móvil, también conocida como aplicación, es un software desarrollado para su uso en cualquier tipo de dispositivo móvil, como un teléfono inteligente o una tableta. Estos tipos de aplicaciones pueden ser aplicaciones independientes o aplicaciones distribuidas en diferentes dispositivos, redes y/o servidores. (Enríquez & Casas, 2013).

Esencialmente, una aplicación sigue siendo software. Por tanto, teóricamente podríamos decir que las aplicaciones móviles son programas para ordenadores, ya sean de sobremesa o portátiles.

En su artículo (Giner, Cetina, Fons & Pelechano, 2010) dijo que “los dispositivos móviles equipados con diferentes tipos de sensores han ayudado a eliminar la barrera existente entre el mundo real y el mundo virtual de los sistemas de información. Los sistemas móviles actuales son conscientes del entorno físico en el que se utilizan y adaptan los servicios que proporcionan en consecuencia. Por ejemplo, al tocar un libro con un dispositivo móvil, podemos obtener información sobre su autor o tomarlo prestado sin hacer cola”.

Las aplicaciones móviles al estar instaladas en un dispositivo electrónico que usamos cotidianamente permiten tener una interacción directa con el propietario de la vivienda, convirtiéndose en una herramienta portátil que proveerá al usuario la posibilidad de gestionar el acceso de los visitantes de manera ágil y precisa.

Las aplicaciones móviles facilitan a los usuarios la compra de sus productos en línea o en la vida real, donde los clientes pueden obtener ofertas especiales ya sean descuentos u ofertas en productos, descuentos y cupones, este tipo de aplicación permite recopilar opiniones y sugerencias de sus clientes para mejorar sus servicios o por el contrario, un cliente muy satisfecho promocionar o recomendar esos otros leads en Facebook,

Instagram, Twitter, etc. Sus redes sociales fomentan la comunicación entre las empresas y sus consumidores. Además, se pueden integrar infinidad de opciones, desde una galería de productos y servicios hasta un formulario de contacto y mapa de ubicación.

Las aplicaciones móviles son programas que se pueden descargar a dispositivos móviles (celulares, tabletas, reproductores de mp3), que brindan diferentes funciones: comunicaciones, ofimática, entretenimiento.

Una aplicación móvil es una aplicación informática que se desarrolla para su ejecución a través de un dispositivo móvil inteligente, una tableta u otro dispositivo móvil en el que el usuario desee ejecutar. Estos se encuentran en las tiendas, a través de los cuales el público es accesible a quienes deseen utilizarlos (García, 2015).

Cada dispositivo móvil según su marca utiliza un sistema operativo (SO) diferente, entre los que más usuarios tienen se encuentran: iOS (Apple), Android (Google) y Windows Phone (Microsoft).

Tipos de aplicaciones móviles

En su investigación (Segovia, 2014) describe que, a nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico.

1. Aplicaciones nativas

Las aplicaciones nativas son aplicaciones que se han desarrollado utilizando software proporcionado por cada sistema operativo a los programadores, comúnmente conocido como Kit de desarrollo de software o SDK. Entonces, Android, iOS y Windows Phone tienen una tras otra y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje que usa el SDK, este tipo de aplicaciones se descargan e instalan desde diferentes plataformas (Delía, Galdamez, Thomas, & Pesado, 2013).

Este tipo de aplicaciones se actualizan continuamente en donde el usuario debe volver a descargarse la aplicación o a su vez seleccionar la pestaña de actualización para que automáticamente se actualice a la última versión del aplicativo en donde en la mayoría de sus actualizaciones se ven las mejoras y depuran los errores que pudo haber existido en esa versión.

Comentado [1]: Indique en algún lugar que va a utilizar de lo que expone que le sirve. por ejemplo al final. Consideramos que el desarrollo nativo es el mejor porque

Comentado [2R1]: _Marked as resolved_

Comentado [3R1]: _Re-opened_
En ese caso Ing, nosotros elegimos las aplicaciones híbridas. Entonces en el apartado de abajo ubicamos porque las consideramos mejor y porque la vamos a utilizar. O ¿También debemos agregar eso que usted nos indica con las nativas? o ya no?. Me indica, gracias!

Este tipo de aplicaciones no requieren Internet ni un plan de datos para su funcionamiento, por lo que brindan una experiencia de usuario más fluida y están realmente integradas en los dispositivos móviles, lo que permite el uso completo de todas las funciones del hardware del terminal, incluidas las cámaras y los sensores. (GPS, acelerómetro, giroscopio, etc.).

Sin embargo, también poseen ciertas desventajas como por ejemplo el hecho de no poder utilizar la misma aplicación en dos o más sistemas operativos distintos, sino que es necesario volver a escribir el código de la aplicación en el lenguaje nativo de cada dispositivo, es decir, se deberá realizar una nueva aplicación para cada sistema operativo siendo tres de los principales: iOS, Android, Windows Phone. Esto sin duda genera costos sumamente altos en la fase de desarrollo del producto software en función de solventar gastos operativos.

A nivel de diseño, esta capa de aplicaciones cuenta con una interfaz basada en las instrucciones de cada sistema operativo, consiguiendo una mayor coherencia y consistencia con el resto de aplicaciones y con el propio sistema operativo. Esto promueve la facilidad de uso y beneficia directamente a los usuarios que encuentran las interfaces familiares.

Criterios	Android	iOS	Windows Phone
Lenguajes	Java, Kotlin	Swift, Objective-C	C#, Visual Basic .NET
Herramientas	Android SDK	XCode	Visual Studio, Windows Phone Dev Tools
Ejecutables	.apk	.app	.xap

*Tabla 2. Principales entornos de desarrollo según el sistema operativo
Fuente: Elaboración propia*

Cabe señalar que en la actualidad existen herramientas de programación tales como Xamarin Forms, NativeScript, Flutter, etc., que integran el SDK de cada sistema operativo permitiendo un desarrollo multiplataforma con la particularidad de poder acceder a todas las características y funcionalidades propias de hardware. Esto es posible debido a que, si bien es cierto, el código de la aplicación está escrito en un

lenguaje distinto, al compilar el proyecto este pasa al lenguaje nativo de cada sistema operativo lo cual nos permite reutilizar código y disminuir exponencialmente el tiempo de desarrollo.

2. WebApps

La base de programación de las aplicaciones web, también conocidas como aplicaciones web, es HTML, así como JavaScript, herramientas ya conocidas por los programadores web.

En este caso no se utiliza el SDK, este es programable independientemente del sistema operativo en el que se utilizará la aplicación. Por ello, estas aplicaciones se pueden utilizar fácilmente en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar código diferente para cada caso concreto.

No es necesario instalar aplicaciones web, ya que se ven con el navegador del teléfono como un sitio web normal como Facebook. Por este motivo, no se distribuyen en la tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente.

Como aplicaciones basadas en la web, los usuarios no necesitan recibir actualizaciones, ya que siempre verán la última versión. Sin embargo, a diferencia de las aplicaciones nativas, requieren una conexión a Internet para funcionar correctamente. (Segovia, 2014)

Las aplicaciones web a menudo tienen una apariencia más genérica independientemente de la apariencia del sistema operativo, por lo que la experiencia de definir al usuario por elementos de navegación e interacción suele ser peor que las aplicaciones nativas.

3. Aplicaciones híbridas

Para el desarrollo de la aplicación móvil, elegimos realizar una aplicación híbrida ya que la codificación de este tipo de apps es multiplataforma y una vez compilado el proyecto es de fácil instalación. Utilizaremos una app para que la parte administrativa del inmueble pueda mostrarles a los demás propietarios de las viviendas todo tipo de información con respecto a eventos sociales o sucesos de interés público que se puedan presentar en la etapa “Murano” de forma inmediata por medio de un sistema de notificaciones. Otra de las funciones ya un tanto más específicas es la posibilidad de

agendar la visita de un familiar o la opción de pedir ayuda inmediata frente a un potencial atentado en contra de la integridad física del usuario y propietario de la vivienda.

Este tipo de aplicaciones integran las mejores características entre las WebApps y las aplicaciones nativas ya que no es necesario aprender el lenguaje de programación nativo, en su defecto se desarrollan con lenguajes de programación web y tecnologías comúnmente conocidas como HTML y CSS. Otra de las ventajas de estas aplicaciones es que no son dependientes a los navegadores web, sino que se instalan de forma similar a las aplicaciones nativas.

Todo esto permite obtener un desarrollo multiplataforma, es decir, con el mismo código se compila aplicaciones tanto para iOS como para Android y en algunos casos más específicos, para la plataforma universal de Windows (UWP) sin ningún tipo de limitante ya que, por medio de librerías, se accede a todas las características de hardware del dispositivo, tal y como en una aplicación nativa.

Visualmente no se puede apreciar diferencias significativas entre una misma aplicación instalada en dos o más sistemas operativos, por lo cual se dice que estas aplicaciones no se identifican con el sistema operativo como en el caso de la aplicación de Netflix que visualmente es igual en Android y en iOS pues en la mayoría de los casos los controles se asemejan a los de Material Design (Cuello & Vittone, 2013).

1.4.4 Tecnologías utilizadas

Lenguajes de programación

Entre los lenguajes de programación de alto nivel utilizados para el desarrollo de nuestro proyecto están:

1. Hypertext Preprocessor (PHP)

Para el desarrollo del servicio web se va a utilizar el lenguaje de programación PHP, hemos seleccionado este lenguaje debido a que posee un poderoso framework de código abierto y con una amplia documentación bajo el patrón de diseño Modelo-Vista-Controlador. Este patrón de diseño provee una estructura bien definida y organizada lo que facilita la reutilización de código y por lo tanto reduce significativamente el lapso en la fase de desarrollo.

PHP (o Preprocesador de Hipertexto) es un lenguaje de programación de alto nivel y de código abierto comúnmente conocido por la comunidad ya que es uno de los más utilizados en el desarrollo web por ser fácil de incrustar en estructuras de código HTML. En lugar de utilizar largos comandos para generar HTML (como en C o en Perl), las páginas de PHP contienen etiquetas con instrucciones propias del lenguaje embebidas en código HTML que al ser ejecutadas por el servidor retornan al lado del cliente alterando la página que se visualizó inicialmente (Napurí, 2021).

2. JavaScript

JavaScript, como todo lenguaje de programación, tiene ciertas peculiaridades como: sintaxis, modelo de objetos, etc. Obviamente, todo distingue un idioma de otro. Además, JavaScript es un lenguaje relativamente peculiar en su enfoque de las cosas. Esta sección es fundamental para cualquier principiante que quiera aprender a programar e incluso para aquellos que ya conocen un lenguaje de programación porque las diferencias con otros lenguajes de programación son muchas.

JavaScript permite programar scripts. Como se mencionó anteriormente, se usa un lenguaje de programación para escribir el código fuente que es analizado por la computadora. Hay tres formas de utilizar el código fuente:

- Lenguaje compilado: El código fuente se da a un programa llamado compilador que lee el código fuente y lo convierte en un lenguaje que el equipo será capaz de interpretar: el lenguaje binario, es de 0 y 1. Lenguajes como C o C++ son los lenguajes compilados más reconocidos.
- Lenguaje pre compilado: aquí el código fuente está parcialmente compilado, generalmente un código que es más fácil de leer por la computadora, pero aún no es un código binario. Este código intermedio se denomina "Máquina Virtual", que ejecutará el código.
- Lenguaje interpretado: en este caso, no hay compilación. El código fuente permanece sin cambios y si desea ejecutarlo, debemos proporcionar un intérprete que leerá y realizará las acciones solicitadas.

Los scripts son en su mayoría interpretados. Y cuando decimos que JavaScript es un lenguaje interpretado, significa que es un lenguaje cuyas instrucciones se van

ejecutando en tiempo real según las acciones que realice el usuario. Por tanto, es necesario contar con un intérprete para ejecutar código JavaScript, y el intérprete que se utiliza con frecuencia se incluye en tu navegador de internet.

Con JavaScript resulta sumamente sencillo capturar los objetos del DOM y manipular sus propiedades para diseñar pantallas mucho más interactivas y así ofrecer una mejor experiencia de usuario.

Cada navegador tiene un intérprete de JavaScript, que varía de un navegador a otro. Si usa Internet Explorer, el intérprete se llama JScript (el intérprete de la versión 9 se llama Chakra), en Mozilla Firefox se llama SpiderMonkey y el motor V8 es de Google Chrome.

Frameworks

Un framework se puede definir como un entorno de trabajo para desarrollar aplicaciones, ya sean web o de escritorio, que proporciona componentes que respaldan el trabajo de los programadores, como bibliotecas, bibliotecas de funciones, plantillas de uso, administración de recursos de tiempo de ejecución y más. Esto permite completar el proyecto sin tener que escribir mucho código, lo que hace que el trabajo sea más eficiente y recursivo (es decir, promueve la reutilización del código).

1. Laravel

Laravel es el nombre de un framework creado para trabajar con PHP en 2011, fundado por Taylor Otwell y con el tiempo se ha vuelto dominante sobre otros marcos para trabajar con PHP como Symfony o Zend Framework.

Es un framework de desarrollo que tiene una curva de aprendizaje muy rápida y gestiona una sintaxis limpia, con el fin de eliminar la complejidad del desarrollo web al facilitar tareas comunes, como autenticación, enrutamiento, sesiones y almacenamiento en caché.

Proporciona las potentes herramientas necesarias para crear potentes aplicaciones y se puede utilizar tanto para empresas como para proyectos más pequeños, lo que significa que es perfecto para todo tipo de proyectos.

Los desarrolladores que elijan usar Laravel al construir sus proyectos recibirán los siguientes beneficios:

- Reducir costos y tiempo durante el desarrollo y posterior mantenimiento de la aplicación creada.
- Curva de aprendizaje relativamente baja en comparación con otros marcos PHP.
- Flexible y adaptable no sólo para usar el sistema MVC tradicional, sino también para reducir las líneas de código que proporciona, a las que llama "Rutas cerradas".
- Modular y con un amplio paquete y sistema de drivers cuya funcionalidad se puede ampliar de forma sencilla, robusta y segura.
- Facilidad de uso de datos gracias a Eloquent, un ORM de interoperabilidad de base de datos totalmente orientado a objetos que es compatible con la gran mayoría de bases de datos del mercado.
- Facilita la gestión de las rutas de nuestra aplicación, así como genera URLs amigables que mejoran el posicionamiento en la web.
- Utiliza el sistema del modelo Blade, que se caracteriza por su sencillez y además incluye un sistema de caché que las hace más rápidas.
- Una gran comunidad y mucha documentación especialmente en su sitio web oficial
- Tiene una herramienta de línea de comandos llamada Artisan que te permite programar tareas programadas como ejecutar migraciones.

Elegimos esta tecnología para el desarrollo del web api a la que se conectará la aplicación móvil y demás transacciones del lado del servidor porque Laravel es un framework de código abierto, fácil de implementar y además cuenta con una amplia documentación tanto en la página oficial como en foros de desarrollo. Otro de los factores que fueron considerados al momento de inclinarnos hacia una tecnología u otra es el Object-relational mapping (ORM), el ORM de Laravel provee una amplia variedad de filtros y objetos Queryset que son una representación del modelo mismo, pero con la característica de que estos son objetos iterables y que por medio de ellos se puede acceder a toda la data de la tabla a la que hacen referencia en el modelo.

2. Vue.js

Vue.js es un nuevo motor de JavaScript creado por Evan You, un miembro muy conocido de la comunidad por su participación en el desarrollo de Meteor y por haber sido desarrollador de Google durante varios años.

Evan You (2019), define su herramienta como un framework progresivo. Progresivo porque el marco se divide en diferentes bibliotecas bien definidas con responsabilidades específicas. De esta forma, el desarrollador integra diferentes módulos según las necesidades del contexto en el que se encuentra. No es necesario incluir todas las funcionalidades en primer lugar como en el caso de frameworks como AngularJS 1.x o EmberJS 1.x.

Es un sistema modular bastante similar al de React. Facebook desarrolló un kernel para poder trabajar con vistas, pero desde entonces se han creado una serie de bibliotecas (tanto de Facebook como de la comunidad) que permiten un trabajo productivo en un SPA. Aquí, todas las partes importantes están enmarcadas en un proyecto Vue.js creado por Evan You.

El núcleo permite el desarrollo de componentes de interfaz de usuario a través de JavaScript. La biblioteca está enmarcada en arquitecturas de componentes (muy de moda) con gestión de modelos internos basada en el patrón MVVM. Esto significa que los componentes, internamente, tienen un mecanismo dual de "enlace de datos" para manipular el estado de la aplicación.

Cuenta con una biblioteca bastante rápida que permite renderizar en mejores tiempos que React. Estos son los tiempos que podemos encontrar en su documentación:

Vue.js	React	Rendimiento
23ms	63ms	Rápido
42ms	81ms	Mediano
51ms	94ms	Promedio
73ms	164ms	95%
343ms	453ms	Lento

*Tabla 3. Comparativa del tiempo que tarda en renderizar Vue.js frente a React
Fuente: (Hanchett & Listwon, 2018)*

Además de esto, VueJS proporciona componentes visuales de forma reactiva. Piezas de UI bien encapsulados que exponen una API con propiedades de entrada y emisión de eventos.

Elegimos utilizar esta tecnología para el desarrollo de todo lo correspondiente al frontend de la aplicación web porque Vue es un potente framework de JavaScript que permite desarrollar aplicaciones de tipo SPA, lo cual se ajusta al tipo de desarrollo que realizamos.

Las single-page application (SPA) son muy utilizadas en la actualidad ya que son más rápidas a comparación de una aplicación web tradicional (MPA), el motivo de que sean más rápidas tiene que ver con la forma en la que se realizan las consultas al servidor, una SPA realiza las consultas desde la vista por medio de Ajax por lo cual se dice que son reactivas (no se necesita recargar la página), mientras que las MPA realizan la transacción del lado del servidor y luego renderizan la información actualizada nuevamente hacia la vista (lo cual produce que se recargue la página). Esto sumado a la posibilidad de acceder fácilmente a todos los objetos del DOM y sus propiedades, fue lo que nos motivó a utilizar esta herramienta.

3. Bootstrap

En los últimos años y con la irrupción de la llamada web 2.0, Internet ha pasado un gran cambio para adaptarse a todas las necesidades de los usuarios y por ello las webs también han tenido que cambiar mucho. Uno de los cambios más significativos es la aparición del concepto de responsive o adaptativo, es decir, diseños que se adaptan al tamaño de pantalla de cualquier dispositivo capaz de navegar por la red.

Esto se puede hacer mediante el uso de técnicas CSS avanzadas o mediante el uso de marcos desarrollados por empresas o terceros. Este es el caso de Bootstrap creado por Twitter. A lo largo de este artículo, hablaremos sobre Bootstrap, uno de los marcos CSS más famosos que utilizan los desarrolladores de portales en la actualidad.

El framework Bootstrap se lanzó en 2011. Originalmente fue desarrollado por Twitter, aunque luego se lanzó bajo licencia del MIT. Hoy sigue creciendo en el repositorio de GitHub. Es un marco que brinda la capacidad de crear un sitio web totalmente receptivo mediante el uso de bibliotecas CSS. En estas bibliotecas podemos

encontrar una gran cantidad de elementos ya desarrollados y listos para usar, como botones, menús, cuadros e incluso una larga lista de fuentes.

Desde su lanzamiento, Bootstrap se ha presentado como una gran herramienta para crear interfaces de usuario limpias que se adaptan por completo a cualquier dispositivo y pantalla, independientemente del tamaño.

Sin duda, una de las características más importantes y útiles que ofrece Bootstrap es el sistema de cuadrícula Fig. 1, donde todos los elementos se pueden escalar a un ancho de 12 columnas de diferentes tamaños según el dispositivo desde el que se visualice la web. Este sistema está diseñado para ayudarnos a maquetar los elementos del sitio web y su adaptación a diferentes tamaños de pantalla.

Elegimos utilizar el framework Bootstrap como un complemento a Vue JS para cierta parte de la aplicación web porque esta tecnología facilita el maquetado de la vista por medio de librerías de CSS y JavaScript. Estas librerías son muy útiles para alinear objetos HTML y definir puntos de ruptura según el tamaño de las pantallas que acceden a la aplicación, al definir puntos de ruptura nos aseguramos de que la aplicación podrá ser utilizada tanto en ordenadores como en dispositivos móviles.



Fig. 1 Descripción gráfica del sistema de rejillas de Bootstrap

Fuente: (Krause, 2016)

4. Quasar Framework

Quasar es un framework basado en Vue.js de código abierto bajo la licencia MIT que permite a los desarrolladores web crear rápidamente sitios web/aplicaciones receptivas de varios tipos:

- SPA (aplicación de una sola página)

- SSR (aplicación renderizada del lado del servidor) (+ adquisición de cliente PWA opcional)
- PWA (aplicación web progresiva)
- BEX (extensión del navegador)
- Aplicaciones móviles (Android, iOS, ...) a través de Córdova o Capacitor
- Aplicaciones de escritorio multiplataforma (usando Electrón).

El lema de Quasar es: codifique una vez e implemente simultáneamente como sitio web, aplicación móvil y/o aplicación Electrón. Sí, una base de código para todos, que lo ayuda a crear aplicaciones en un tiempo récord utilizando una interfaz de línea de comandos moderna y con la tecnología de componentes web Quasar súper rápidos y el cumplimiento de las mejores prácticas.

Elegimos utilizar el Quasar framework para el desarrollo de la aplicación móvil debido a que con esta herramienta podemos generar código que será compatible con los dos sistemas operativos líderes del mercado, es decir, Android y iOS.

Con el Quasar framework desarrollaremos la app que servirá como puerta de acceso a todos los servicios de la aplicación web. El motivo por el cual no se especifican otras tecnologías para desarrollo móvil es porque toda la lógica de negocio estará ubicada en el servidor web del cual obtendremos y manipularemos la información desde la aplicación móvil por medio de APIs.

Base de Datos

Las bases de datos son el método preferido para almacenar datos estructurados. Desde grandes aplicaciones multiusuario hasta teléfonos móviles y organizaciones electrónicas, utilizan tecnología de bases de datos para garantizar la integridad de los datos y facilitar el trabajo de los usuarios y los programadores que las desarrollaron.

Al definir una "base de datos", es importante tener claro qué son los datos y la información porque estos elementos son esenciales para el crecimiento de la base de datos, según (Juárez, 2006, p.45):

- Dato: es un conjunto de caracteres con un significado determinado, pueden ser numéricos, alfabéticos o alfanuméricos, es la unidad mínima de información. Los datos en la base de datos responden a la función (objeto, propiedad, valor).
- Información: Conjunto de datos ordenados que se manejan de acuerdo a las necesidades del usuario, para que un conjunto de datos sea procesado eficientemente y se genere información, primero se debe almacenar adecuadamente en archivos.

Para definir “base de datos” tenemos que volver al archivo y lo que es la tupla, pues según (Arreguin, 2006, p. 9) el archivo se interpreta como un conjunto de datos interdependientes, recopilados, que satisfacen las necesidades de información de una determinada comunidad de usuarios, que han creado muchas bases de datos.

Según (Gómez, 2007, p.18) una base de datos es un conjunto de datos pertenecientes a un mismo contexto, almacenados sistemáticamente para su uso posterior, es un conjunto de datos estructurados en un modelo que refleja las relaciones y restricciones forzadas a existir en el mundo real. Los datos deben ser compartidos por diferentes usuarios y las aplicaciones deben ser independientes de ellos, y la definición y descripción de esos datos debe ser única, almacenada con ellos.

Por otro lado (Juárez, 2006, p.45) establece que una base de datos es una colección de datos almacenados entre ellos que tienen relaciones lógicas y están diseñados para satisfacer las necesidades de información de una empresa u organización.

1. PostgreSQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos estandarizado utilizado por la gran mayoría de los servidores de bases de datos que administran bases de datos relacionales o relacionales de objetos.

Es un lenguaje declarativo en el que los comandos especifican cuál debe ser el resultado, no cómo lograrlo (como en los lenguajes de procedimiento). Al ser de carácter declarativo, es muy sistemático, sencillo y con un aprendizaje muy agradable ya que sus palabras claves logran que puedas escribir comandos como si fueran frases que definen (en inglés) lo que quieres obtener (Momjian, 2001).

Para definir la base de datos que se utilizará en nuestro proyecto consideramos tanto el volumen de datos que se pretenden gestionar como la cantidad de usuarios simultáneos que potencialmente accederán a la aplicación web. Con base en la revisión de la literatura fue posible determinar que PostgreSQL es la mejor opción entre las bases de datos relacionales de código abierto ya que provee características orientadas al modelo de desarrollo que aplicamos. Una de estas características es que cuenta con un mejor manejo de concurrencia, lo cual permite que varios procesos puedan manipular datos de forma simultánea.

CAPÍTULO 2

2. METODOLOGÍA

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta en cada una de las actividades del ciclo de vida de un proyecto de desarrollo con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea, basándose en una combinación de modelos de proceso genéricos, definiendo artefactos como roles y actividades, junto con prácticas y técnicas recomendadas que se usan para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos.

Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos.

2.1 Desarrollo de Software con Metodología en Cascada

El primer modelo relacionado en la industria del software fue el Modelo de Cascada, llamado así por la ubicación de las etapas en su desarrollo, que parece caer "por gravedad" a las siguientes fases. Fue propuesto en 1970, por el investigador Winston Royce, es una metodología que ordena estrictamente las fases del proceso de desarrollo de software, por lo que el inicio de cada fase debe esperar a la finalización del proceso de desarrollo de software. (Prieto Álvarez, 2015)

Según el investigador Winston Royce, impulsor de este modelo, el beneficio de esta metodología surge cuando no existe una fecha de implementación inmediata, por lo que se requiere un tiempo adecuado para desarrollar cada etapa. Es importante mencionar que para que este modelo tenga un índice de riesgo más bajo, los requisitos deben ser muy claros y haber sido establecidos formalmente en la primera parte del proyecto. (Prieto Álvarez, 2015)

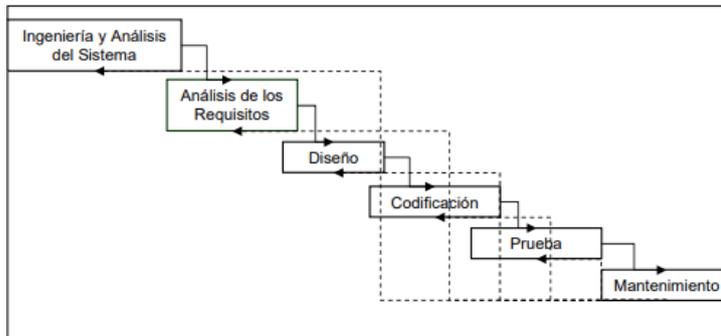


Fig. 2 Modelo de Cascada

Fuente: Elaboración propia

2.1.1. Fases del Modelo de Cascada

Fase de Análisis de Requerimientos

Todo proyecto de software comienza con una fase de análisis que incluye un estudio de viabilidad y la definición de requisitos. En un estudio de factibilidad se evalúa el costo, la rentabilidad y la factibilidad del proyecto de software. El estudio de viabilidad proporciona las especificaciones del proyecto (descripción general de las necesidades), el plan y las estimaciones financieras del proyecto, así como las propuestas al cliente, si es necesario. (Galindo Aguilar et al., 2010)

Luego, se da una definición detallada de los requisitos, incluyendo un análisis del escenario de salida y un concepto. Si bien el análisis de salida es responsable de describir el problema en sí, el concepto debe definir las funciones y características que el producto de software debe proporcionar para cumplir con los requisitos respectivos. La definición de los requisitos conduce a una especificación, una descripción detallada de cómo se deben implementar los requisitos del proyecto, así como un plan para las pruebas de aceptación, entre otros.

Fase de Diseño

La fase de diseño sirve para formar una solución específica basada en los requisitos, tareas y estrategias identificadas en la fase anterior. En esta fase, los desarrolladores de software son responsables del diseño de la arquitectura del software, así como del anteproyecto, centrándose en componentes específicos, como las interfaces, el lugar de trabajo o la biblioteca. La fase de diseño conduce a un bosquejo aproximado del plan de

diseño del software, así como planes de prueba para varios componentes. (Galindo Aguilar et al., 2010)

Fase de Codificación

La arquitectura de software se diseña en la fase de diseño que se implementa en la fase de codificación, que incluye la programación de software, la depuración y las pruebas unitarias. Durante esta fase, el proyecto de software se convierte en el lenguaje de programación elegido. Los diferentes componentes se desarrollan por separado, se prueban a través de pruebas unitarias y se integran gradualmente en el producto final. La fase de codificación conduce a que un producto de software se pruebe primero como producto final en la siguiente fase (prueba alfa). (Galindo Aguilar et al., 2010)

Fase de Pruebas

La fase de prueba incluye la incorporación del software en el entorno seleccionado. Por regla general, los productos de software se entregan primero a usuarios finales seleccionados durante las pruebas beta. Las pruebas de aceptación se desarrollan durante la fase de análisis que ayuda a determinar si el software cumple con los requisitos previamente definidos. Los productos de software que superan las pruebas beta están listos para su lanzamiento. (Galindo Aguilar et al., 2010)

Fase de Mantenimiento

Una vez superada con éxito la fase de prueba, se permite la aplicación efectiva del software. La etapa final del modelo en cascada incluye la distribución, mantenimiento y mejora del software.

CAPÍTULO 3

3. PROPUESTA DE SOLUCIÓN

3.1 Tema

Análisis, diseño e implementación de una aplicación móvil y un sistema web para gestionar el ingreso de visitas a la urbanización “La Gema”.

3.2 Descripción de la propuesta

Nuestra propuesta consiste en el desarrollo de un sistema web y una aplicación móvil para el control de acceso a la urbanización “La Gema”.

En base al problema planteado y al análisis del mismo se desarrolló una aplicación móvil donde solo el propietario de la vivienda puede autorizar el ingreso de los visitantes a través de un nuevo sistema de control de seguridad, el cual permite colocar el nombre del visitante y la hora creando un código QR el cual puede ser compartido por WhatsApp y que servirá para el acceso a la urbanización.

Así mismo, para brindar una mejor administración en la urbanización se desarrollará una aplicación Web que permita llevar el control de los pagos de las alcúotas. Brindando todas las opciones para realizar una administración eficiente y poder respuestas a los habitantes de la urbanización; evitando así conflictos entre el personal administrativo y los residentes.

La aplicación web y móvil, automatiza los procesos que actualmente se realizan de forma manual. Además, gracias a las aplicaciones se obtendrán resultados eficientes y confiables.

3.3. Desarrollo de Software con Metodología en Cascada

Unas de las ventajas de esta metodología es que cada etapa tiene un conjunto de metas bien definidas y las actividades dentro de cada una contribuyen a la satisfacción de metas de esa fase o quizás a una su secuencia de metas de la misma.

3.1.1. Fases de la Metodología en Cascada

Fase de Análisis de los requerimientos

Con la finalidad de comprender las problemáticas que existen en la Urbanización “La Gema”, se realizó la recolección de información mediante una entrevista dirigida al Ing.

Roberto León presidente de los residentes, para definir las necesidades existentes, además determinar las características y funcionalidades de la aplicación móvil y Sistema Web a realizar con la finalidad de alcanzar los objetivos planteados.

Obtenida la información a través de la entrevista, se diseñó una base de conocimiento con la información de los elementos que se han utilizado para la creación de aplicaciones móviles que gestionen el ingreso en las urbanizaciones, así como sistemas de administración de alcótuas. Lo que ayudó a definir los recursos necesarios para la creación del sistema móvil y web, por ejemplo; el lenguaje de programación, framework, base de datos, herramientas de diseño, prototipos, librerías y api.

Fase de Diseño

Se descompone y organiza el sistema por separado según los requerimientos obtenidos por medio de la entrevista, creando una arquitectura completa del sistema a desarrollar. Así realizando los algoritmos necesarios cumpliendo los requerimientos del usuario, también ver los análisis necesarios para que herramienta usar en la etapa de codificación.

Una vez finalizada la fase de requerimientos se procedió a la fase de diseño, en donde se realiza un prototipo general de la arquitectura, la misma que deberá desenvolverse en interacciones progresivas hasta conseguir transformarse en el software final, dicho de otra manera, se encargó de detallar la estructura interna de la página web y de qué manera está relacionada con las entidades que lo complementan.

A diferencia de los modelos en la fase de análisis que representan los requerimientos del usuario, los modelos utilizados en esta fase tratan de las características del sistema que permitirán implementarlo de manera efectiva, entre las características, sus módulos deben estar a cargo de una tarea específica. y estar debidamente relacionados entre sí, lo que ayudará a que el mantenimiento del sistema sea mucho más fácil.

Es decir que en esta fase se experimentan las posibles opciones de implementación de la aplicación web a construir y se definirá la estructura general que tendrá, además de utilizar una base de datos empleando un diagrama de relación de entidades.

La construcción de la base de datos es uno de los pasos principales dentro del desarrollo de la aplicación, para este fin el modelo físico se lo generó con la ayuda de software de terceros, el cual contiene el detalle específico de la estructura del sistema y la descripción de cada una de sus partes, así como las relaciones existentes.

Se definió una estructura de base datos con su respectivo diagrama del Modelo Entidad – Relación (E-R), que permitirá determinar cómo será el funcionamiento de los procesos del aplicativo, ya que estos diagramas nos ayudarán en la construcción del software.

A continuación, se presentan los diagramas realizados:

Diagrama de caso de uso

Diagrama del caso de uso de administrador

En este diagrama podemos observar el actor administrador de la urbanización que tendrá tres roles principales que serán el de registrar los residentes con sus respectivas viviendas y familiares, generar usuario para los residentes y verificar el pago de alcúotas de manera general.

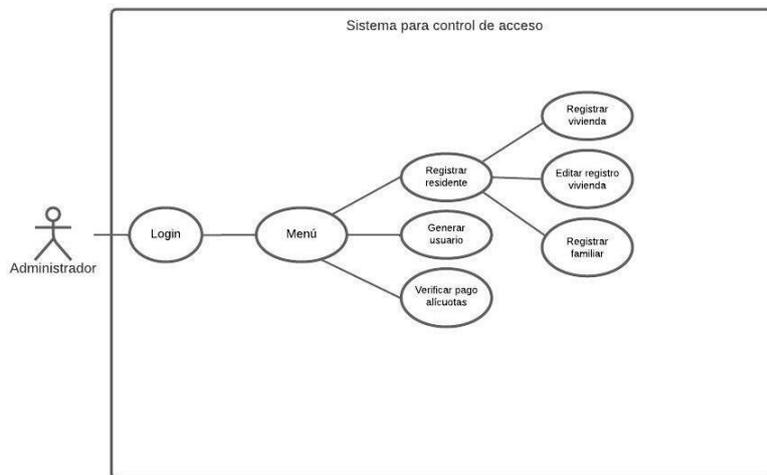


Fig. 3 Diagrama de casos de uso del administrador

Fuente: Elaboración propia

Diagrama de caso de uso residente

En este diagrama podemos observar el actor residente de la urbanización que tendrá tres funciones principales que son la de registrar un nuevo visitante, generar invitación y compartir invitación.

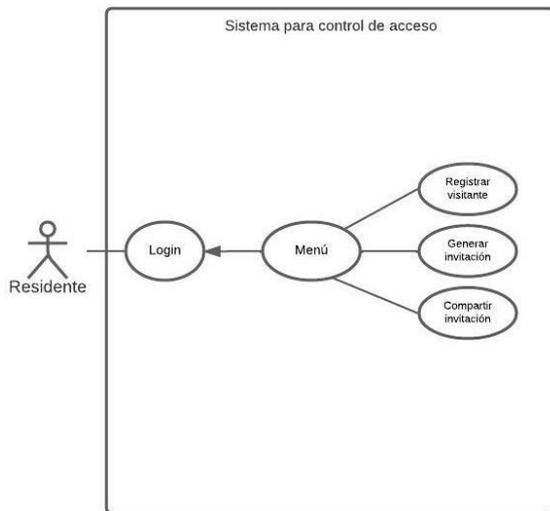


Fig. 4 Diagrama de casos de uso del administrador

Fuente: Elaboración propia

Diagrama de entidad/relación

Diagrama del modelo entidad relación

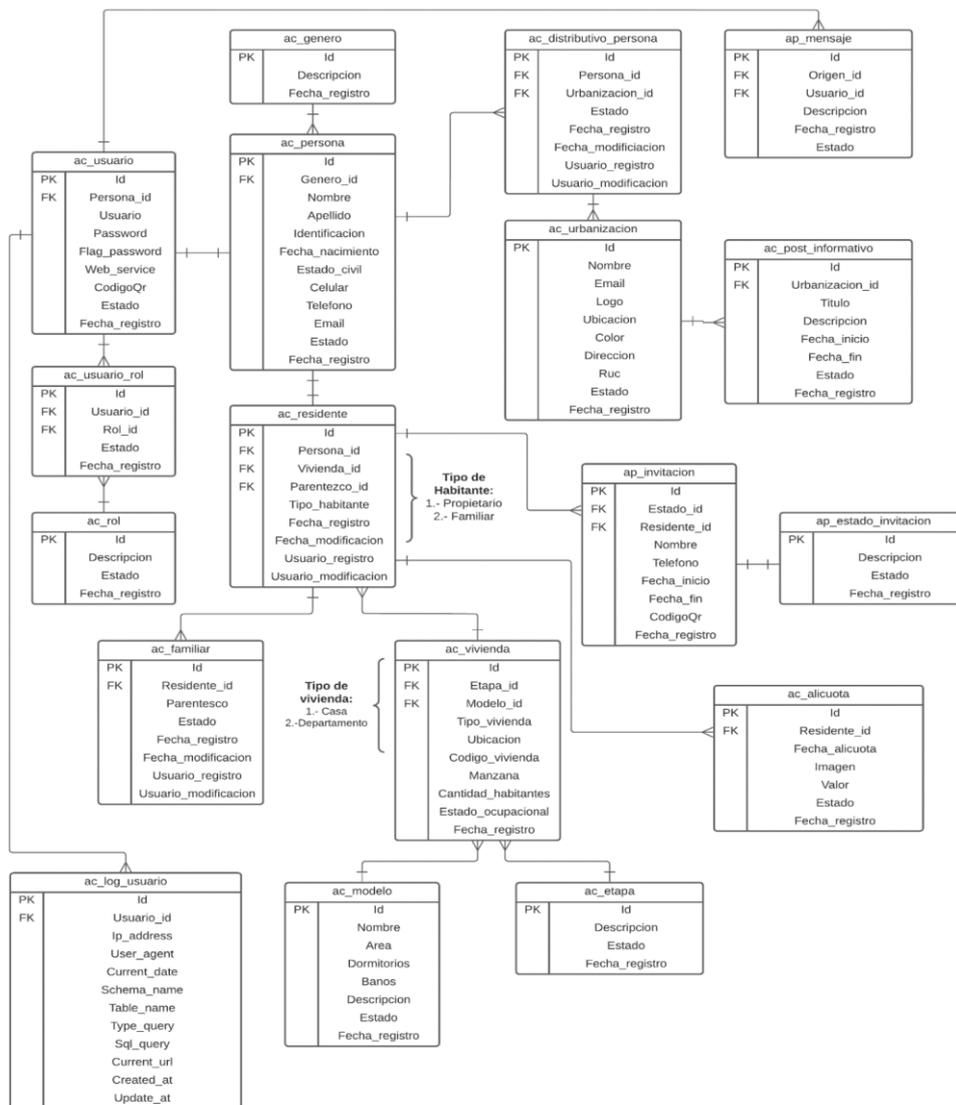


Fig. 5 Modelo entidad/relación de la aplicación web

Fuente: Elaboración propia

Descripción de las tablas implementadas

Tabla Género

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria de género
E	descripción	Integer	Descripción del género: 1-Masculino, 2-Femenino
E	fecha_registro	Datetime	Fecha en la que se registró un nuevo género

Tabla 4. Descripción de la tabla de Género
Fuente: Elaboración propia

Tabla Usuario

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria de usuario
FK	persona_id	Integer	Clave foránea para identificar a la persona.
E	usuario	Varchar(50)	Usuario que usará la persona para acceder al sistema
E	Clave	Varchar(50)	Contraseña que usará la persona para acceder al sistema
E	flag_password	Varchar(50)	Bandera que se usará para la contraseña
E	web_service	Varchar(50)	Servicio web
E	codigoqr	Varchar(50)	Ruta en la que se almacenará la imagen del código QR

E	estado	Boolean	Estado para habilitar e inhabilitar usuario.
E	fecha_registro	Datetime	Fecha en la que se registró un nuevo usuario

*Tabla 5. Descripción de la tabla Usuario
Fuente: Elaboración Propia*

Tabla Usuario Rol

Tipo	Atributo	Tipo de Dato	Descripción
PK	id	Integer	Identificador único o clave primaria
FK	usuario_id	Integer	Clave foránea para identificar al usuario.
FK	rol_id	Varchar(50)	Clave foránea para identificar el rol.
E	estado	Boolean	Estado para habilitar e inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro.

*Tabla 6. Descripción de la tabla roles de usuario
Fuente: Elaboración Propia*

Tabla Rol

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Identificador único o clave primaria de la tabla.
E	descripción	Integer	Descripción del rol.
E	Estado	Boolean	Estado para habilitar e inhabilitar un registro.

E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro.
---	----------------	----------	---

*Tabla 7. Descripción de la tabla rol
Fuente: Elaboración propia*

Tabla Log Usuario

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Identificador único o clave primaria de la tabla.
FK	usuario_id	Integer	Clave foránea que identifica al usuario.
E	ip_address	Varchar(20)	Estado para habilitar e inhabilitar un registro.
E	user_agent	Datetime	Fecha en la que se insertó un nuevo registro.
E	current_date	Datetime	La fecha actual (al momento de insertar un registro)
E	schema_name	Varchar(50)	Nombre del esquema de la base de datos.
E	table_name	Varchar(50)	Nombre de la tabla que se afectó
E	type_query	Varchar(50)	Tipo de consulta
E	sql_query	Varchar(100)	Sentencia SQL que se ejecutó desde el sistema
E	current_url	Varchar(80)	Dirección URL actual
E	created_at	Datetime	Fecha en la que se insertó un nuevo registro.
E	update_at	Datetime	Fecha en la que se modificó un nuevo registro.

*Tabla 8. Descripción de la tabla Log de usuarios
Fuente: Elaboración propia*

Tabla Persona

Tipo	Atributo	Tipo de Dato	Descripción
PK	id	Integer	Identificador único o clave primaria de la tabla.
FK	urbanizacion_id	Integer	Clave foránea que identifica la urbanización
FK	genero_id	Integer	Clave foránea para identificar el género de la persona
E	nombre	Varchar(50)	Nombres completos de la persona.
E	apellido	Varchar(50)	Apellido paterno y materno de la persona
E	identificacion	Varchar(10)	Cédula o pasaporte
E	fecha_nacimiento	Datetime	Fecha de nacimiento
E	estado_civil	Integer	Estado civil: 1-Soltero, 2-Casado
E	celular	Varchar(10)	Teléfono celular
E	telefono	Varchar(20)	Teléfono convencional
E	email	Varchar(20)	Dirección de correo electrónico
E	estado	Boolean	Estado para habilitar e inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro.

*Tabla 9. Descripción de la tabla persona
Fuente: Elaboración propia*

Tabla Residente

Tipo	Atributo	Tipo de Dato	Descripción
-------------	-----------------	---------------------	--------------------

PK	Id	Integer	Identificador único o clave primaria de la tabla residente.
FK	persona_id	Integer	Clave foránea para identificar a la persona.
FK	vivienda_id	Integer	Clave foránea para relacionar a la persona
FK	parentesco_id	Integer	Clave foránea para relacionar al familiar
E	tipo_habitante	Varchar(50)	Tipo de habitante: 1-Propietario, 2-Familiar
E	estado	Boolean	Estado para habilitar e inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro
E	fecha_modificacion	Datetime	Fecha en la que se modificó un registro
E	usuario_registro	Varchar(20)	Usuario que creó el registro para auditoría
E	usuario_modificacion	Varchar(20)	Usuario que modificó un registro

*Tabla 10. Descripción de la tabla residente
Fuente: Elaboración propia*

Tabla Familiar

Tipo	Atributo	Tipo de Dato	Descripción
PK	id	Integer	Identificador único o clave primaria de la tabla residente.

FK	residente_id	Integer	Clave foránea para identificar a la persona.
E	parentesco	Varchar(50)	Descripción del parentesco
E	estado	Boolean	Estado para habilitar e inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro
E	fecha_modificacion	Datetime	Fecha en la que se modifica un registro
E	usuario_registro	Varchar(20)	Usuario que creó el registro
E	usuario_modificacion	Varchar(20)	Usuario que modificó un registro

*Tabla 11. Descripción de la tabla familiar
Fuente: Elaboración propia*

Tabla Distributivo Persona

Tipo	Atributo	Tipo de Dato	Descripción
PK	id	Integer	Identificador único o clave primaria de la tabla residente.
FK	persona_id	Integer	Clave foránea para identificar a la persona.
FK	urbanizacion_id	Integer	Clave foránea para relacionar a la persona
E	estado	Boolean	Estado para habilitar e inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se insertó un nuevo registro

E	fecha_modificacion	Datetime	Fecha en la que se modifica un registro
E	usuario_registro	Varchar(20)	Usuario que creó el registro para auditoría
E	usuario_modificacion	Varchar(20)	Usuario que modificó un registro

*Tabla 12. Descripción de la tabla Distributivo de persona
Fuente: Elaboración propia*

Tabla Ac_vivienda

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica a la vivienda
FK	Etapa_id	Integer	Clave foránea para identificar a la persona
FK	Modelo_id	Integer	Clave foránea para identificar el modelo de la vivienda
E	Tipo_vivienda	Integer	Identificar los dos tipos de viviendas existentes: 1.- Departamento; 2.- Casa
E	Ubicación	Varchar	Ubicación de la vivienda
E	Codigo_vivienda	Varchar	Código de identificación de la vivienda
E	Manzana	Varchar	Manzana de Ubicación de la vivienda
E	Cantidad_habitantes	Integer	Cantidad de personas que viven en la vivienda
E	Estado_ocupacional	Integer	Estado de la vivienda

E	Fecha registro	DateTime	Guardar la fecha y hora del registro
---	----------------	----------	--------------------------------------

*Tabla 13. Descripción de la tabla vivienda
Fuente: Elaboración propia*

Tabla Ac_urbanizacion

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica a la urbanización
E	Nombre	Varchar	Nombre de la urbanización
E	email	Varchar	Email institucional de la Urbanización
E	Logo	Varchar	Logo de la Urbanización+
E	Ubicación	Varchar	Ubicación de la urbanización
E	Ruc	Varchar	Ruc corporativo de la Urbanización
E	Color	Varchar	Color que identifique a la Urbanización
E	Dirección	Varchar	Dirección principal de la urbanización
E	Estado	Integer	Estado de la urbanización
E	Fecha registro	DateTime	Fecha y hora del registro

Tabla 14. Descripción de la tabla Urbanización

Fuente: Elaboración propia

Tabla Ac_modelo

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica el modelo de la vivienda
E	Nombre	Varchar	Nombre del modelo
E	Area	Double	Área de construcción
E	Dormitorio	Integer	Cantidad de dormitorios
E	Banos	Integer	Cantidad de Baños
E	Descripción	Varchar	Descripción del modelo
E	Estado	Integer	Estado del modelo
E	Fecha registro	DateTime	Fecha y hora del registro

Tabla 15. Descripción de la tabla Modelo de vivienda
Fuente: Elaboración propia

Tabla Ac_alicutas

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica a la alícuota
FK	Residente_id	Integer	Clave foránea para identificar al residente
E	Fecha_alicuota	Date	Fecha del pago de Alícuota
E	Imagen	Varchar	Imagen del comprobante de pago
E	Valor	Double	Valor mensual de la Alícuota

E	Estado	Integer	Estado 1.- Pagado; 2.- Atrasado; 3.- Pendiente
E	Fecha_registro	DateTime	Fecha y hora del registro

*Tabla 16. Descripción de la tabla Alicuota
Fuente: Elaboración propia*

Tabla Ac_post_informativo

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica al post informativo
FK	Urbanizacion_id	Integer	Clave foránea para identificar a la Urbanización
E	Titulo	Varchar	Título del Post
E	Descripcion	Varchar	Descripción del Post
E	Imagen	Varchar	Imagen del Post informativo
E	Fecha_inicio	Date	Fecha de inicio que se mostrará el post
E	Fecha_fin	Date	Fecha hasta que se mostrará el post
E	Estado	Integer	Estado del post

E	Fecha_registro	DateTime	Fecha y hora de Registro
---	----------------	----------	--------------------------

*Tabla 17. Descripción de la tabla Post de información
Fuente: Elaboración propia*

Tabla Ap_mensaje

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica el mensaje
FK	Origen_id	Integer	Clave foránea para identificar el usuario que envía el mensaje
FK	Usuario_id	Integer	Clave foránea para identificar el usuario que recibe el mensaje
E	Descripcion	Varchar	Descripción del Mensaje
E	Fecha_registro	DateTime	Fecha y hora de Registro
E	Estado	Integer	Estado del mensaje

*Tabla 18. Descripción de la tabla Mensaje de la App
Fuente: Elaboración propia*

Tabla Ap_invitacion

Tipo	Atributo	Tipo de Dato	Descripción
------	----------	--------------	-------------

PK	Id	Integer	Clave primaria con valor único que identifica la invitación
FK	Estado_id	Integer	Clave foránea para identificar el estado que se encuentra la invitación
FK	Residente_id	Integer	Clave foránea para identificar el residente que crea la invitación
E	Nombre	Varchar	Nombre de la persona que recibe la invitación
E	Telefono	Varchar	Teléfono de la persona que recibe la invitación
E	Fecha_inicio	DateTime	Fecha y hora de inicio de validación de la invitación
E	Fecha_fin	DateTime	Fecha y hora de fin de validación de la invitación
E	CodigoQr	Varchar	Imagen del código Qr generado
E	Fecha_registro	DateTime	Fecha de creación del registro

Tabla 19. Descripción de la tabla invitación de la App
Fuente: Elaboración propia

Tabla Ap_estado_invitacion

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria con valor único que identifica estado de la invitación
E	Descripcion	Varchar	Descripción del Estado
E	Estado	Integer	Estado del registro
E	Fecha_Registro	DateTime	Fecha y hora de registro

Tabla 20. Descripción de la tabla Estado de la invitación de la App
Fuente: Elaboración propia

Tabla Etapa

Tipo	Atributo	Tipo de Dato	Descripción
PK	Id	Integer	Clave primaria de la tabla ac_etapa
E	Descripción	Integer	Descripción de la etapa
E	Estado	Boolean	Este campo se utiliza para habilitar o inhabilitar un registro.
E	fecha_registro	Datetime	Fecha en la que se registró una nueva etapa de la urbanización

Tabla 21 Descripción de la tabla etapa de la aplicación web

Fuente: Elaboración propia

Diseño del Sistema Web

Login

La Fig. 4 corresponde a la pantalla del Login. Esta es la pantalla inicial de nuestro sistema, es decir, cuando el usuario acceda a la url <http://18.209.55.95/> en su navegador web se visualizará esta pantalla en donde se le pedirá ingresar su usuario y contraseña. En caso de presentar algún inconveniente con sus credenciales el usuario deberá contactar con un administrador de La Gema.

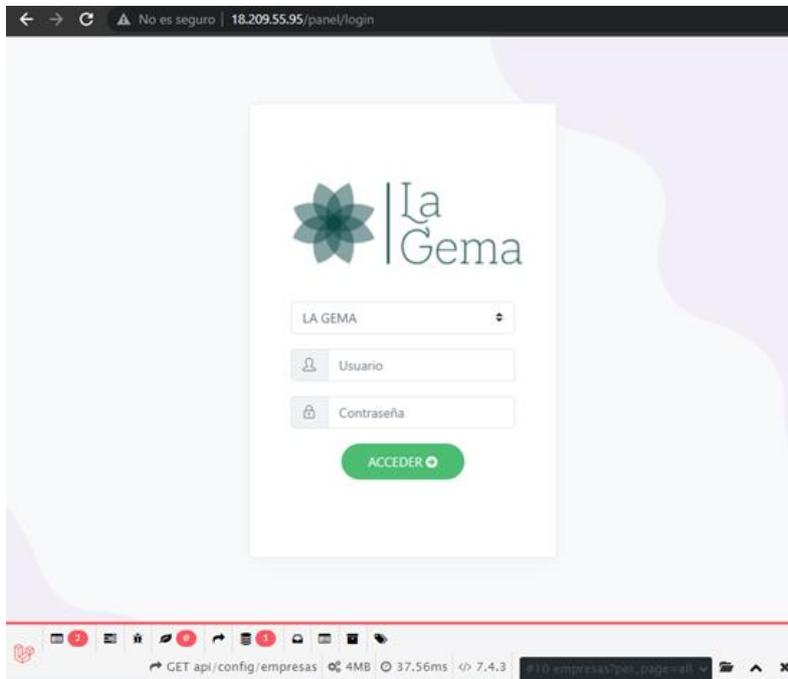


Fig. 6 Pantalla de Login

Fuente: Elaboración propia

Menú principal

Esta pantalla es la que se visualizará luego de iniciar sesión. En la parte central se encuentra el módulo acceso, que contiene las funcionalidades principales de la aplicación como se muestra en la Fig 6. Al lado derecho bajo el logo se visualizarán los anuncios creados en la vista de la Fig 9. de manera opcional según los requerimientos del administrador.

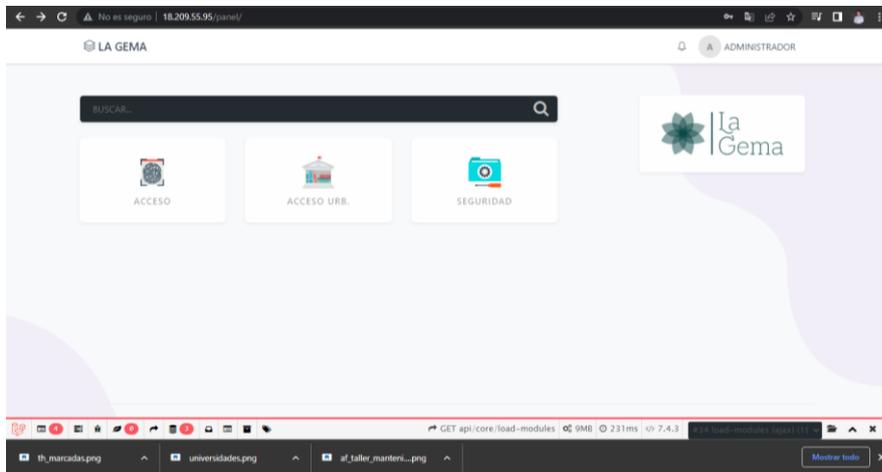


Fig. 7 Menú de la aplicación

Fuente: Elaboración propia

Listado de alcúotas.

En esta pantalla se puede visualizar el listado mensual de alcúotas en donde podrá buscar por nombre del residente, además la opción de agregar alcúota que le permitirá al administrador generar los rubros que deben cancelar los residentes de cada mes y las opciones de editar y eliminar alcúota.

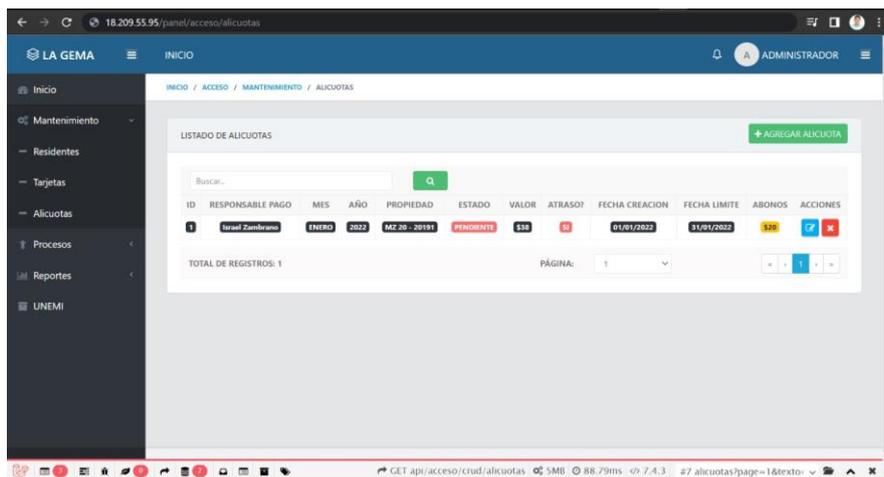


Fig. 8 Listado de alcúotas

Fuente: Elaboración propia

Mantenimiento de residentes

En esta pantalla se puede visualizar el formulario de ingreso de residentes, dentro de los campos a ingresar se incluye inicialmente la identificación que puede ser cédula o pasaporte. Solo en el caso de ingresar un pasaporte extranjero se le desplegarán en pantalla los campos nombres, apellidos y fecha de nacimiento. En el caso de ingresar la cédula los nombres, apellidos y fecha de nacimiento se almacenarán directamente. Por último, al pulsar el botón guardar se presentará un listado con los residentes ingresados.

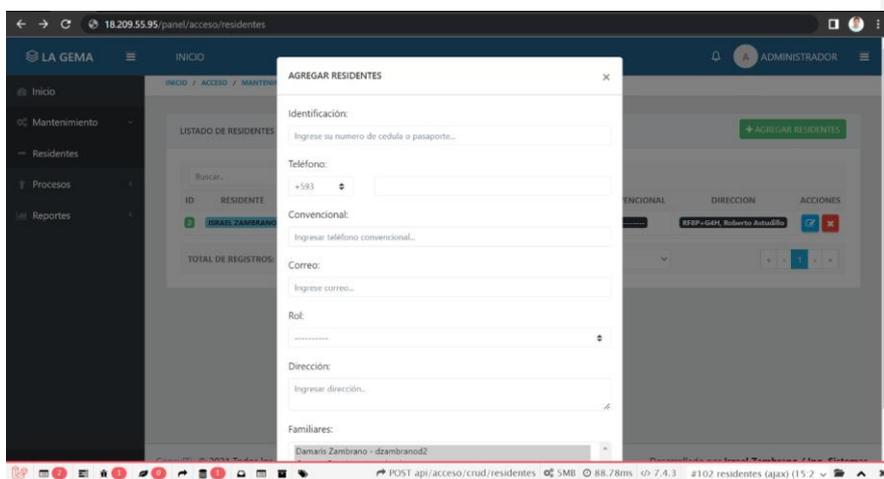


Fig. 9 Pantalla para mantenimiento de residentes

Fuente: Elaboración propia

Listado de tarjetas

En esta pantalla se visualizará el listado de todos los anuncios que el administrador necesite comunicar a los residentes en forma de tarjetas que se desplegarán dentro de una ventana emergente al ingresar a la aplicación móvil. Además, se muestran las opciones agregar, editar y eliminar tarjeta.

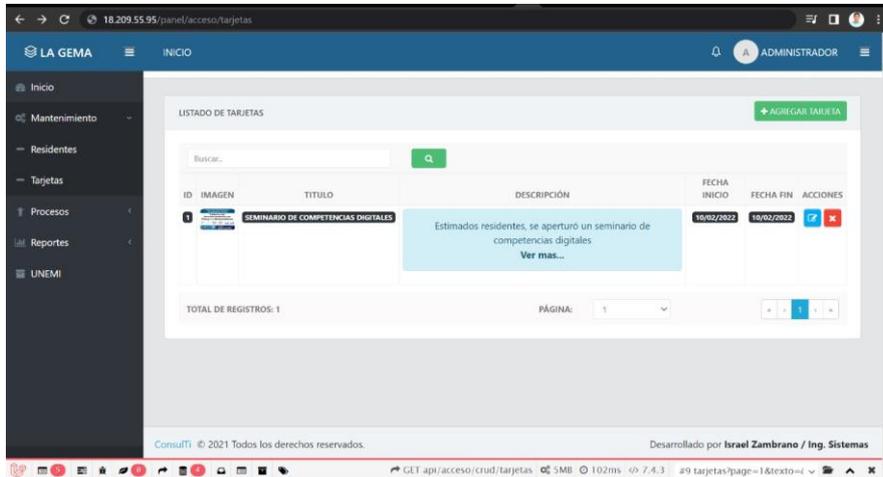


Fig. 10 Pantalla para listado de tarjetas

Fuente: Elaboración propia

Agregar Tarjeta

En esta pantalla se puede visualizar el formulario para registrar una nueva tarjeta en donde los campos título, imagen, fecha inicio y fecha fin son de carácter obligatorio.

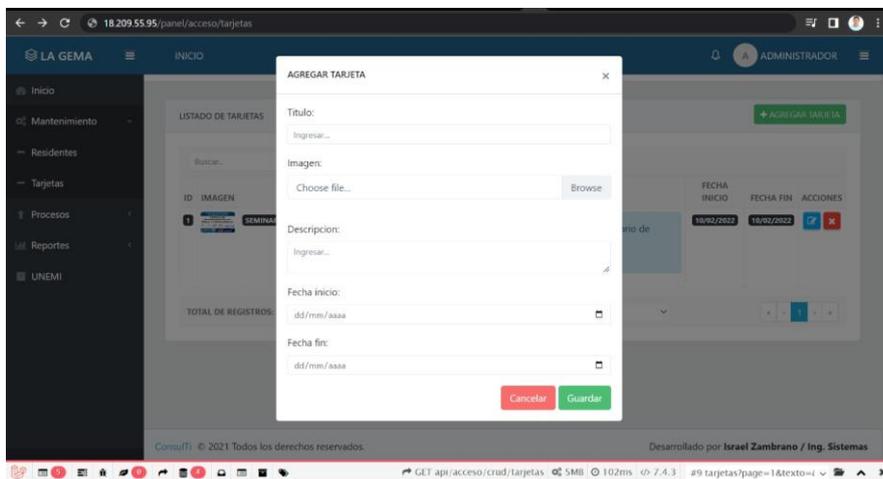


Fig. 11 Formulario para agregar una nueva tarjeta.

Fuente: Elaboración propia

Listado de urbanizaciones

En esta pantalla se puede visualizar el listado de urbanizaciones o etapas de las mismas, esto con el fin de que la aplicación web sea escalable y fácil de comercializar. En el listado se muestra un único registro que realizamos los autores, el botón de agregar una nueva urbanización, editar y eliminar.

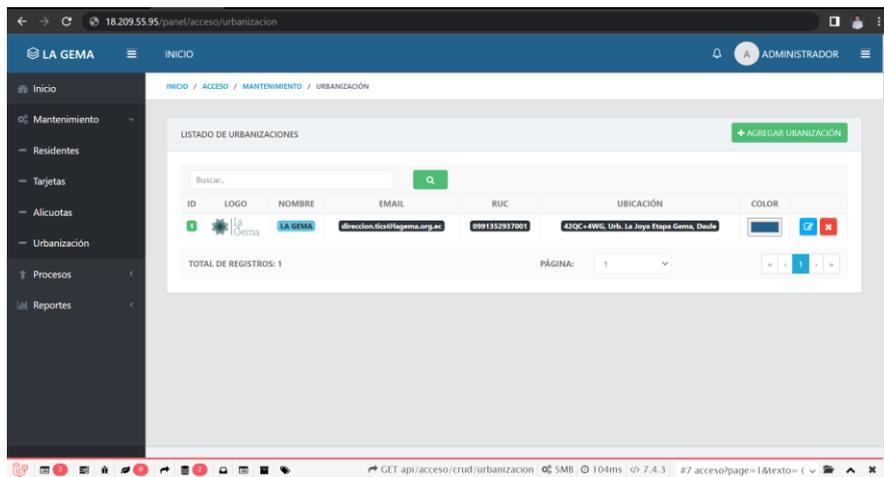


Fig. 12 Listado de urbanizaciones.

Fuente: Elaboración propia

Diseño de la Aplicación Móvil

Inicio de la App

Al momento de abrir la aplicación se mostrará la primera pantalla de inicialización y la segunda pantalla de presentación de la aplicación a través de un carousel.

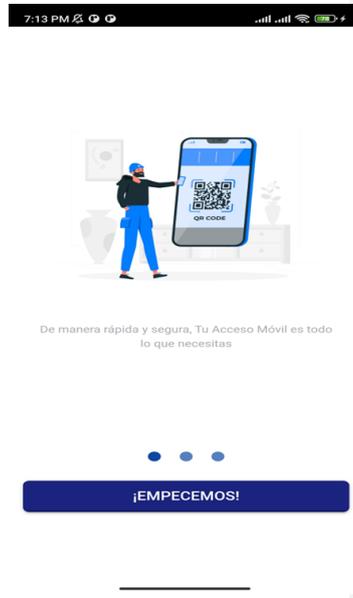


Fig. 13 Inicio de la App

Fuente: Elaboración propia

Login de Acceso a la aplicación

Pantalla de ingreso a la aplicación, siempre y cuando el administrador de la urbanización ya haya creado las credenciales de acceso al usuario que va a ingresar.



Fig. 14 Login de acceso a la aplicación

Fuente: Elaboración propia

Sección de Noticias

En esta pantalla el usuario podrá ver los posts de noticias que publica la Urbanización desde el sistema web. Es muy importante que ayudará a que los residentes estén informados de las últimas novedades de la administración.



Fig. 15 Sección de Noticias

Fuente: Elaboración propia

Pantalla de Inicio

En esta sección se mostrará las invitaciones que tiene activas en ese día. La finalidad es que el residente tenga constancia de las visitas que va a tener en el día.

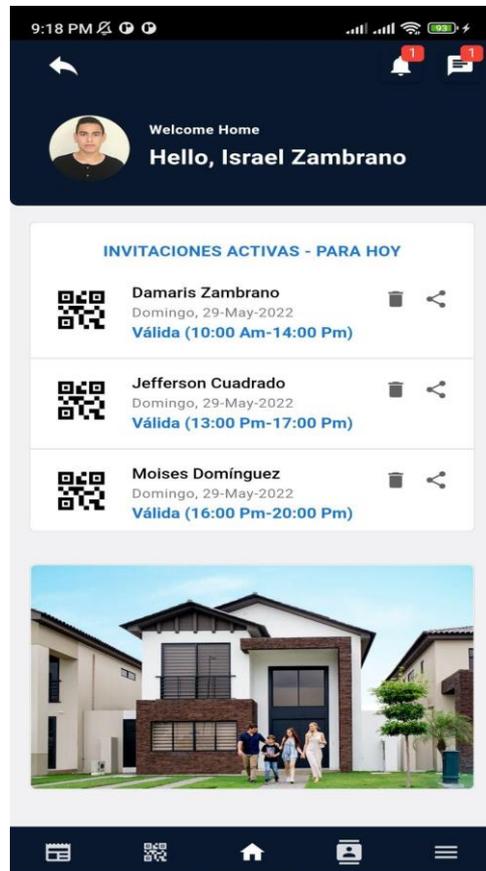


Fig. 16 Pantalla de Inicio

Fuente: Elaboración propia

Listado de Invitaciones

En esta pantalla se mostrarán todas las invitaciones creadas, con sus diferentes estados. Además, cuenta con la opción de compartir invitación, suspender y también crear invitación.



Fig. 17 Listado de Invitaciones

Fuente: Elaboración propia

Compartir Invitación

Se podrá compartir la invitación con su código QR por los diferentes medios nativos del móvil. Además, ayudará a que la persona pueda reenviar el código siempre y cuando esté activa la invitación.



Fig. 18 Compartir Invitación
Fuente: Elaboración propia

Crear Invitación

Pantalla con su respectivo formulario de creación de la invitación, cabe recordar que todos los campos son obligatorios. Además, también tendrá la opción de guardar el contacto siempre y cuando no esté registrado.

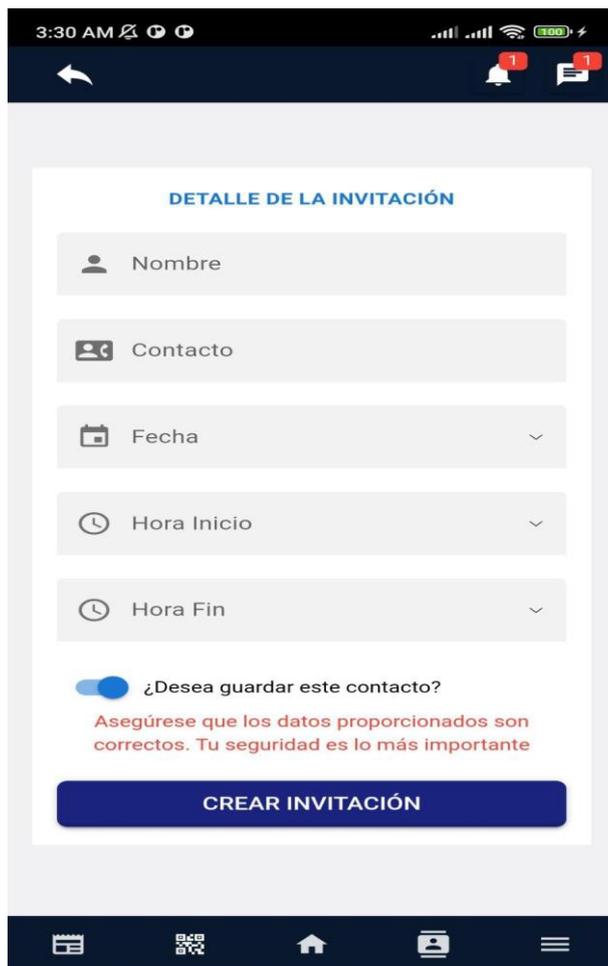


Fig. 19 Crear Invitación

Fuente: Elaboración propia

Contactos Guardados

En esta pantalla el usuario tendrá todos los contactos que ha guardado, además ayudará a la creación de invitaciones de una forma más rápida, debido a que desde el botón más podrá ingresar directamente a la invitación con los datos ya registrados del contacto a invitar.



Fig. 20 Contactos Guardados

Fuente: Elaboración propia

Pantalla de Mensajes

En esta pantalla el usuario podrá comunicarse con la directiva de la urbanización y con los guardias en caso de una emergencia o solicitud.

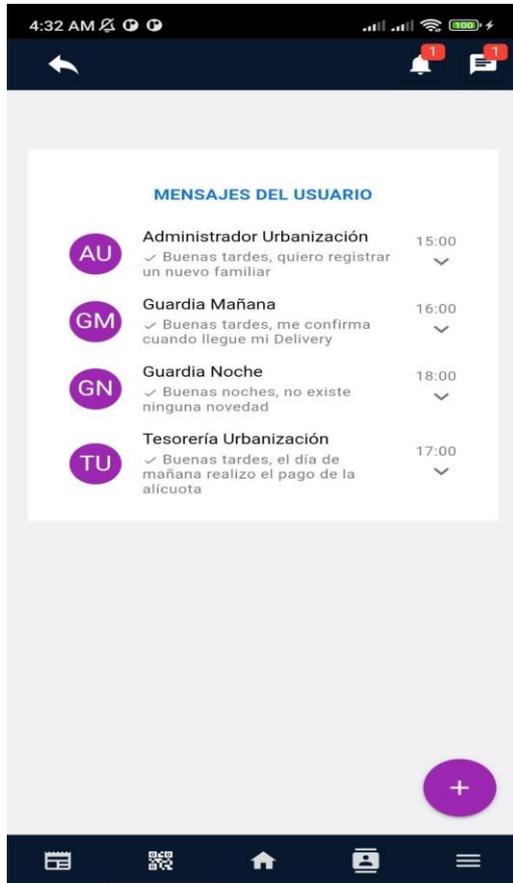


Fig. 21 Pantalla de Mensajes

Fuente: Elaboración propia

Pantalla de Perfil

En esta pantalla el usuario podrá acceder a sus datos personales, su código QR de acceso a la urbanización y las políticas de privacidad de la aplicación.

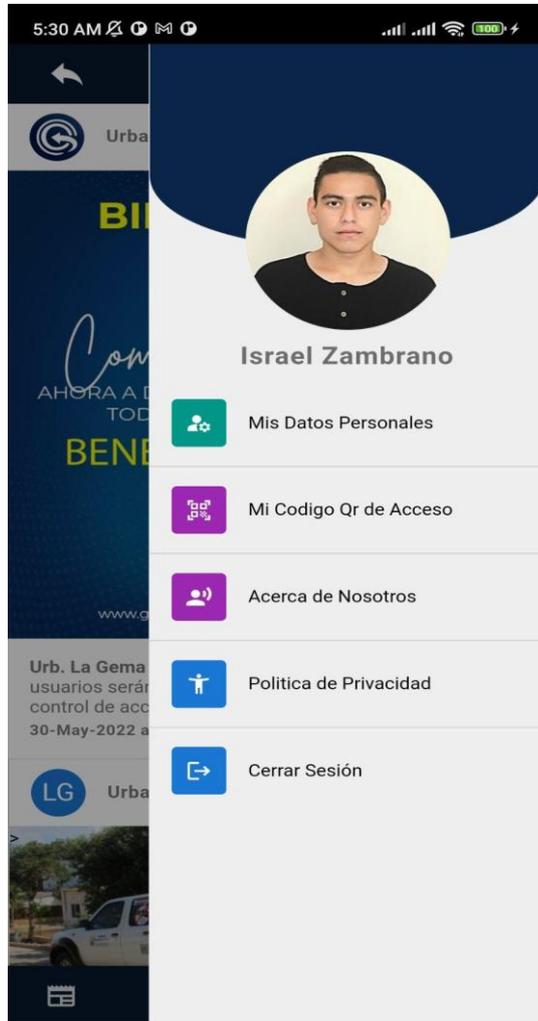


Fig. 22 Pantalla de Perfil

Fuente: Elaboración propia

Fase de Codificación

Esta fase hace referencia a la programación propia del sistema, se desarrolla el código fuente haciendo uso de los prototipos, así como pruebas o ensayos para corregir errores y que cumplan con las especificaciones del diseño. Ya que dependiendo del lenguaje de

algoritmos y su versión, se va creando componentes dentro del mismo proyecto para hacer que la programación sea un proceso más rápido y eficiente.

Una vez realizada las dos fases de análisis y diseño, se empezó el desarrollo tangible del software por medio de los casos de usos que fueron implementados, en esta tercera fase se pueden realizar variaciones en caso de ser necesario ya que el sistema está en desarrollo y refinamiento, una vez finalizado este mini proceso de mejora se inicia la programación de códigos para conseguir un fragmento de lo que será el producto final.

Para el desarrollo del sistema se utilizó Visual Studio Code en su versión 1.67.1 como IDE (Entorno de Desarrollo Integrado), posteriormente se usó PHP en su versión 7.4 como lenguaje de programación, ya que este permite la programación orientada a objetos, lo que nos ayudó para la creación de Apis para el aplicativo móvil en el Framework Quasar y para crear una conexión con Laravel de manera fácil y eficiente. Posteriormente a esto, se usó Laravel mix en su versión 7.5 como base para el desarrollo de nuestra aplicación web, ya que permitió integrar el backend con tecnologías web como: HTML y Vue.js con librerías de Node js, las cuales fueron utilizadas para el desarrollo del FrontEnd, resultando ser uno de los mejores frameworks de conectividad y extensibilidad. Mientras que para el aplicativo Móvil se utilizó el Framework Quasar en su versión 2.6.2 y Vue.js en su versión 3 para el desarrollo del FrontEnd, además se instaló los paquetes de CLI con el marco de desarrollo de aplicaciones móviles Cordova para el empaquetado de código para IOS y Android.

Por todo esto, es que la etapa de desarrollo se considera como una de las fases más complejas y exigentes a lo largo del proyecto, ya que se debe trabajar de manera muy cuidadosa en cada uno de los apartados, es necesario aplicar todos los conocimientos que se han adquirido y así conseguir el desarrollo del producto final de manera exitosa.

A continuación, se muestran algunos fragmentos de código de la aplicación web:

Controlador Urbanización

En este fragmento de código se muestra la creación del api que será consumida desde el template. La función index devuelve una lista de todos los registros en la tabla ac_urbanizacion, en el where le pasamos una variable que contiene la cadena de caracteres que queremos buscar posteriormente desde la vue table. El método store lo utilizamos para almacenar un nuevo registro, con update actualizamos un registro y utilizamos destroy para eliminar.

```
<?php
namespace App\Modules\Acceso\Http\Controllers\crud;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;
use App\Modules\Acceso\Models\crud\Urbanizacion;

class UrbzanizacionController extends Controller
{
    public function __construct(){
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $texto = isset($request->texto) ? strtoupper($request->texto) :
null;
        return Urbanizacion::where('nombre', 'like', '%' . $texto . '%')
            ->isActive()
            ->orderBy('id', 'asc')->VuePaginate();
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
```

```

        //dd($request);
        $request->validate([
            "ruc" => "required|unique:acceso.ac_urbanizacion,ruc",
            "nombre" => "required",
            "email" => "required",
            "logo" => "required",
            "color" => "required",
            "Direccion" => "required",
            "Ubicacion" => "required",
        ]);
        Urbanizacion::create($request->all());
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $request->validate([
            "cedula" => "required|unique:acceso.ac_urbanizacion,ruc, ".$id,
            "nombre" => "required",
            "email" => "required",
            "logo" => "required",
            "color" => "required",
            "Direccion" => "required",
            "Ubicacion" => "required",
        ]);
        Urbanizacion::findOrFail($id)->update($request->all());
    }

    /**
     * Remove the specified resource from storage.
     *

```

```

    * @param int $id
    * @return \Illuminate\Http\Response
    */
    public function destroy($id)
    {
        $cliente = Urbanizacion::findOrFail($id);
        $cliente->estado = 0;
        $cliente->save();
    }
}

```

Consumo de la API desde el template

Como se muestra en este fragmento de código a través de Vue consumimos el api creada anteriormente en el controlador de la urbanización.

```

save_data: function() {
    if (this.type == "store") {
        this.store_data();
    } else {
        this.update_data();
    }
},
store_data: function() {
    this.form
        .post("/api/acceso/crud/urbanizacion")
        .then(({ data }) => {
            this.cancel_modal();
            this.$refs.CoreTable.vueTable();
            this.$message("store");
        });
},
update_data: function() {
    this.form
        .put("/api/acceso/crud/urbanizacion/" + this.form.id)
        .then(({ data }) => {
            this.cancel_modal();
            this.$refs.CoreTable.vueTable();
            this.$message("update");
        });
},
cancel_modal: function() {
    this.$refs.FormModal.hide();
    this.clear_form();
},

```

Fase de Pruebas

Se procede a integrar todas las partes de los subsistemas que funcionan correctamente para formar el sistema completo y así cumplir con todos los requerimientos del software antes de ser puesto en explotación. La prueba se centra en la lógica interna del software asegurando que la entrada definida produce los resultados que se requieren realmente.

El usuario puede acceder a una prueba de la funcionalidad de la página web a través de un servidor de prueba <http://18.209.55.95/panel> en donde se encontrará alojada sistema web. Donde el usuario administrador de la Urbanización puede ir probando la funcionalidad de cada módulo, como son la búsqueda y creación de usuarios, reporte de alcuotas de los residentes por mes, historial de visitas y creación de anuncios que saldrán en el aplicativo móvil.

Mientras que para las pruebas de funcionalidad del aplicativo móvil se le pasará al usuario un archivo Apk para la instalación en su dispositivo móvil. En donde podrá ingresar con sus credenciales creadas por el administrador web y podrá probar la funcionalidad tanto para el rol de Guardia y residentes donde tendrán los módulos de creación de visitas programadas con validación de código Qr, configuración de perfil y añadir personas a la lista negra (es decir prohibido el ingreso a la urbanización).

Una vez comprobado que los aplicativos cumplan efectivamente con cada requisito establecido al principio, se garantiza que el ingreso a la Urbanización por los visitantes se realizará de forma ágil y segura.

Fase de Mantenimiento

Esta fase no se aplica en este proyecto ya que se trata solo de un prototipo.

CONCLUSIONES

De acuerdo a lo realizado en el proyecto de titulación, se han definido algunas terminaciones:

- Concerniente al desarrollo de la aplicación móvil en IOS y Android se lo definió bajo un sistema operativo libre, de manera que el usuario pueda adquirir de manera fácil y rápida la aplicación, se puede concluir que es mucho más fácil de implementar en estos sistemas operativos debido a que se utilizó paquetes de CLI con el marco de desarrollo de aplicaciones móviles Córdova que tienen alta compatibilidad, entre otras ventajas.
- De acuerdo a lo realizado, podemos concluir que la aplicación móvil reducirá la pérdida de tiempo, los riesgos delictivos y la aglomeración de visitantes. Debido a que los familiares o personas externas contarán con la aplicación de fácil uso que permitirá que agenden su visita y acceden fácilmente por medio de un código de validación QR.
- En cuanto a la aplicación Web, los administradores de la urbanización llevarán un control y monitoreo adecuado acerca de las personas que ingresan y salen de la urbanización, así como les facilitará conocer las visitas en tiempo real y evitar pérdida de tiempo, de documentos, entre otras cosas.
- También es importante mencionar que tanto la aplicación móvil como el sistema Web cuentan con una interfaz de fácil uso y amigable con el usuario. De manera que la experiencia del usuario será amena, podrá acceder a la información que requiera y realizar los procesos de visitas de forma rápida y sencilla.

RECOMENDACIONES

En base a la propuesta del proyecto de titulación, se han definido las siguientes recomendaciones:

- En primer lugar, se recomienda capacitar a los responsables de garita que son quienes manejan el sistema Web de visitas en la urbanización para evitar errores de ingreso, actualización y/o eliminación de información, ya que estos se verán reflejados en la aplicación y pueden confundir a los usuarios.
- En la actualidad, los datos son susceptibles a virus informáticos, datos borrados, pérdida de información accidentalmente, entre otros problemas informáticos, es por ello que se recomienda hacer una copia de seguridad de forma periódica en la base de datos.
- La aplicación a futuro tiene una alta probabilidad de ser mejorada, siendo factible que se incluya métodos de pagos por tarjeta en el pago de alcúotas. Ya que actualmente la aplicación móvil permite que los usuarios suban sus comprobantes de pago y el administrador por medio del sistema Web realice la validación correspondiente.

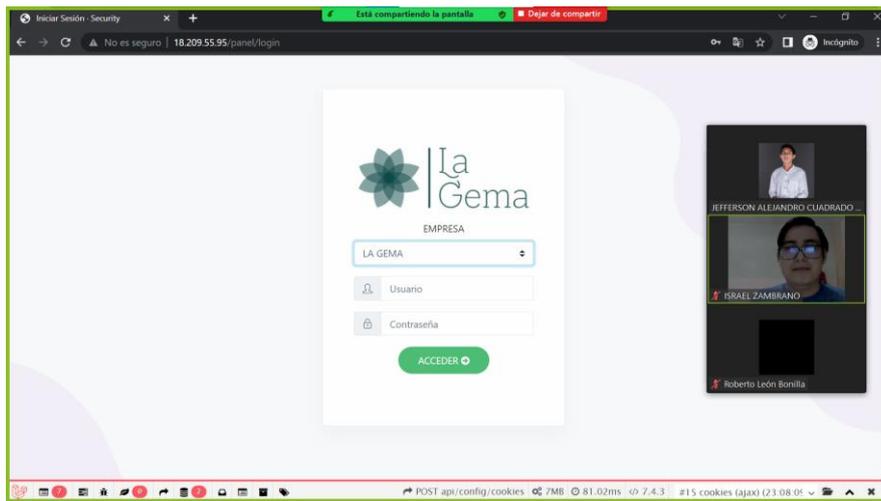
REFERENCIAS BIBLIOGRÁFICAS

- Ahmad, H. W.-U.-D. (2017). *Building Restful Web Services with PHP 7*. Packt Publishing.
- Chimbo, B. (2019). *Diseño de un sistema de administración en urbanizaciones de la ciudad de milagro*. Guayaquil: Universidad de Guayaquil.
- Correa, K., & Castro, J. (2016). *Desarrollo e Implementación de una aplicación para la administración de la Urbanización La Joya, etapa Murano y control de alcuotas*. Guayaquil: Universidad Politécnica Salesiana. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/12306/1/UPS-GT001636.pdf>
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles* (Vol. 1.1). (C. D. Giraldo, Ed.) ePub. Recuperado el 1 de Abril de 2022, de https://app.gumroad.com/read/87cf3834b5960c1b5394be7ecf336341/Zff2xzgzm5kw_cpfEkoelQ==
- Delía, L. N., Galdamez, N., Thomas, P. J., & Pesado, P. M. (2013). Un análisis experimental de tipo de aplicaciones para dispositivos móviles. *XVIII Congreso Argentino de Ciencias de la Computación.*, 766-776.
- Enriquez, J. G., & Casas, S. I. (2013). *Usabilidad en aplicaciones móviles*. Informes Científicos Técnicos-UNPA.
- García, M. (1 de julio de 2015). Investigación de usos y tipos de aplicaciones móviles. *Investigación*, 20-24.
- Giner, P., Cetina, C., Fons, J., & Pelechano, V. (2010). Developing Mobile Workflow Support in the Internet of Things. *IEEE Pervasive Comput*, 9(2), 18-26.
- Hanchett, E., & Listwon, B. (2018). *Vue.js in Action*. Simon and Schuster.
- Krause, J. (2016). *Introduction to bootstrap 4*. Springer.
- Laborde, R., & Paredes, L. (2020). *Implementación de un dispositivo de acceso de control inteligente de gestión de visitantes de una urbanización privada mediante una aplicación móvil usando recursos no renovables*. Guayaquil: Universidad de Guayaquil.
- Marini, E. (2012). El modelo cliente/servidor. Obtenido de <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
- Momjian, B. (2001). *PostgreSQL: introduction and concepts* (Vol. 192). New York: Addison-Wesley.
- Napurí, J. (6 de Septiembre de 2021). *Manual de PHP*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Paredes, C., & Millanes, S. (2020). *Aplicaciones web*. Barcelona: Síntesis.

- Pradilla, E., & Márquez, L. (2016). Los territorios latinoamericanos en la mundialización del capital. *Territorios*, 1(34), 17-34. Obtenido de <http://dx.doi.org/10.12804/territ34.2016.01>
- Solano, J. (2011). Lenguajes de Programación. *Temario*. Lima: Universidad Nacional de Ingeniería, Facultad de Ciencias, 50-54.
- Talón, E. M. (2012). *Apache*. (M. d. Educación, Ed.) Aula Mentor.
- Torres, E. (septiembre de 2015). Estudio del fraccionamiento urbano - social producido por urbanizaciones cerradas y conjuntos privados en la ciudad de Quito: caso "urb. El condado". 50-57. Obtenido de <http://www.dspace.uce.edu.ec/bitstream/25000/9312/1/T-UCE-0001-0234.pdf>
- Galindo Aguilar, I., García Valdovino, R., & Martínez Aguilar, E. (2010). Sistema Web para la gestión de capacitación de la Casa de Cultura Frida Kahlo. México. Obtenido de <https://tesis.ipn.mx/bitstream/handle/123456789/7478/IF7.100.pdf?sequence=1&isAllowed=y>
- Paredes Colmenar, M., & Millanes Santos, J. (2020). *Aplicaciones Web*. Madrid: Juan Carlos Moreno Pérez. Obtenido de <https://www.sintesis.com/data/indices/9788491714729.pdf>
- San Emeterio Villalaín, J. (Julio de 2019). Desarrollo de servicios web para consulta de datos en entornos médicos. *Universidad Politécnica de Cartagena*. Obtenido de <https://repositorio.upct.es/bitstream/handle/10317/7970/tfm-san-des.pdf;jsessionid=35AC2ACAE44E723ACCBDE2A1B566404A?sequence=1>

ANEXOS

Presentación de la aplicación web al administrador de “La Gema”





APP SECURITY

MANUAL DEL USUARIO

Versión: 1.0

Fecha: 11/06/2022

INSTALACIÓN DE LA APP

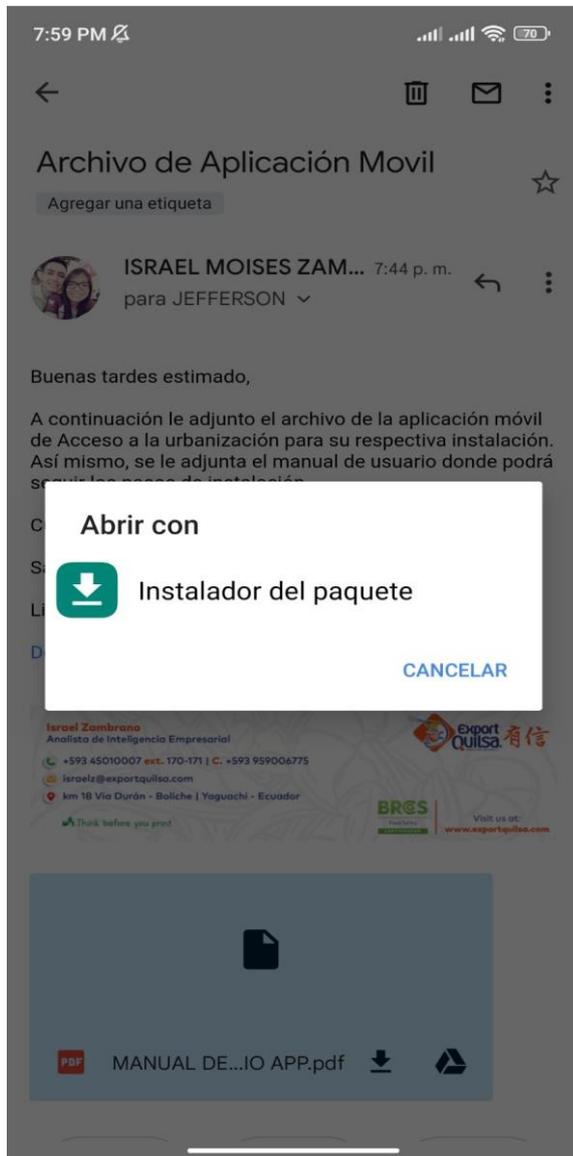
1. Solicitar la creación del usuario de residente al administrador de la urbanización

El administrador será el encargado de crear las credenciales las mismas que llegaran a través de un correo electrónico. Así mismo a través de correo se compartirá el archivo APK para la instalación de dispositivos Android.

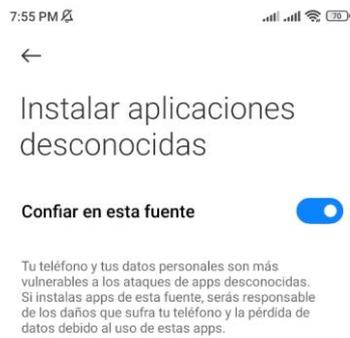
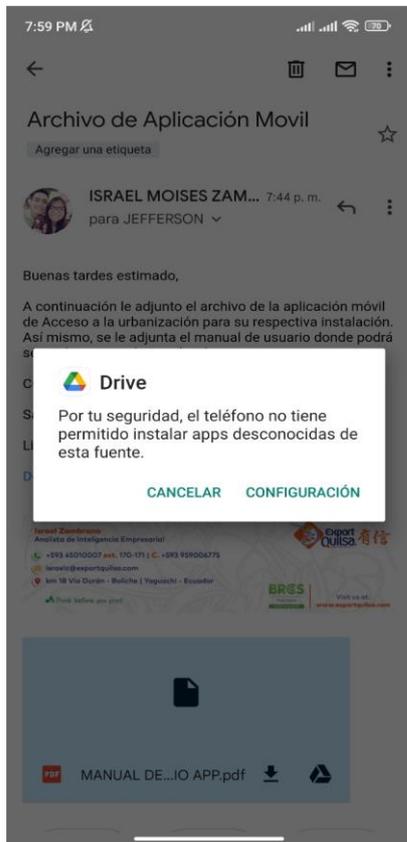


2. Instalación de la aplicación

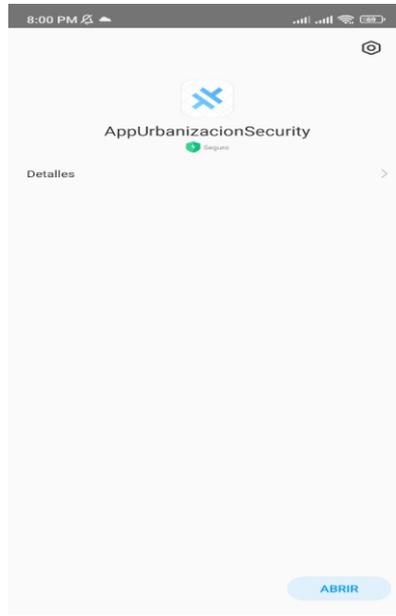
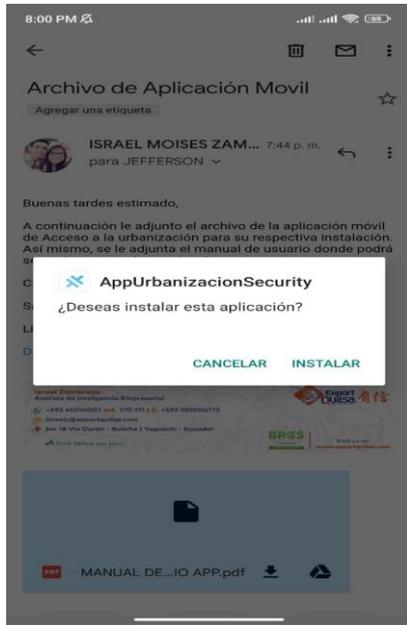
Al momento de hacer click seleccione instalador de paquetes de Android



Le va a solicitar que debe habilitar la opción de instalación de aplicaciones de orígenes desconocidos, desde los ajustes del dispositivo. Ya que es una instalación fuera de Google Play. Da click en configuración y habilite esa opción.



Seleccione en Instalar y espere a que el sistema instale los paquetes de la app. Una vez que ya este instalada le va a salir la opción de abrir aplicación, dele click y comience a disfrutar de las opciones de nuestra aplicación.



1. Inicio de la Aplicación

Para comenzar a utilizar la aplicación es necesario contra con un usuario y contraseña, las cuales serán proporcionada por la urbanización al momento de registrar el residente.





Bienvenido

Para comenzar inicie sesión con su usuario y contraseña

Usuario
izambranod

Contraseña

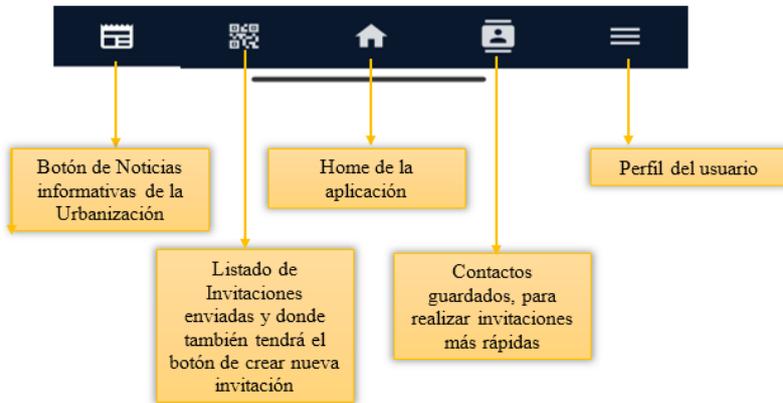
[Forgot Password?](#)

SIGN IN

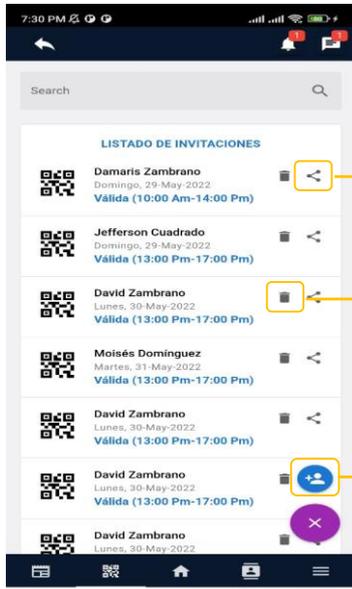
Debe contar con las credenciales proporcionadas por la urbanización

Botón de Inicio de Sesión

1. Menú de la Aplicación



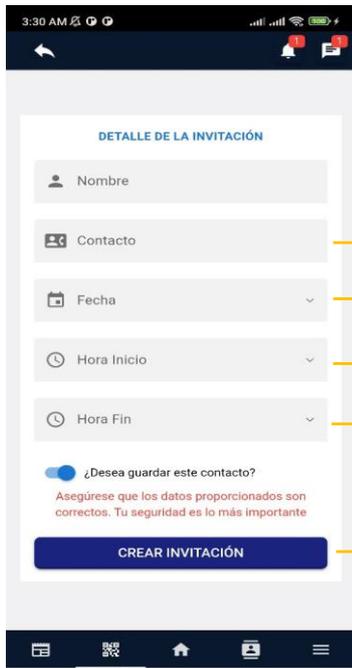
2. Invitación de acceso a la Urbanización



Compartir Invitación, por los diferentes medios que permita el dispositivo.

Cancelar Invitación

Botón que nos manda a la creación de invitación



Celular de la persona que se va a invitar

Fecha que será la invitación

Hora de inicio permitida para el acceso

Hora fin permitida para el acceso

Botón para crear la invitación, cabe recalcar que todos los campos son obligatorios. La invitación solo será válida para la fecha programada.



Compartir la invitación del código Qr, a través de los diferentes medios que permita el dispositivo. La imagen del código Qr será validada solo una vez.

1. Acceso al Código Qr del residente

El residente tiene ya un código de acceso Qr único, es decir que el lo podrá utilizar cada vez que ingrese a la Urbanización.



Código Qr del residente

Cerrar sesión