**VICTUS Residencias**

En este documento se da un primer vistazo a la propuesta de VICTUS residencias, que es un producto administrativo orientado a facilitar la administración de espacios comunes en lugares residenciales con administración activa.

Contexto

Para las personas que viven en parques o conjuntos residenciales, es común encontrar de manera activa, sistemas de administración de reservas con papel, siendo estos poco prácticos y con una facilidad alta a presentar errores

**1.** **¿Por qué estoy aquí?**

**a.** **Juan Daniel Rodriguez Giraldo:** Para programar en el entorno de desarrollo InteliJ y en la interfaz del proyecto, valor agregado: creativo y novedoso, Desventaja: Tiene muchas obligaciones, tanto laborales como académicas.

**b.** **Juan Pablo Avendaño Duque:** El dueño de la idea principal y propósito y con los contactos posibles a donde se puede aplicar, y conocimiento de la comunidad objetivo del proyecto.

**c.** **Andres Felipe Velez Alcaraz:** Para encargarme del diseño y desarrollo del software también coordinando tareas del equipo y asegurando que se cumplan los plazos y objetivos.

**d.** **Luis Miguel :** Encargado de coordinar y asegurar que se cumplan los plazos y objetivos, además de ayudar con las diversas tareas del equipo aplicando los conocimientos adquiridos.

**2.** **Elevator pitch**

**a.** **Para:** Es un proyecto enfocado para la administración de espacios públicos en zonas residenciales, que presenta una administración activa.

**b.** **Que tiene:** La necesidad de una organización mayor y método simple de acceso a la información necesaria para la administración de espacios comunes dentro de una zona residencial administrada, la cual tiene cupos máximos, que a falta de administración puede generar conflictos a la hora del uso compartido de estas zonas.

**c.** **Nuestro producto**

i. **Nombre:** VICTUS Residencias

**d.** **Es un:** Producto de tipo administrativo.

**e.** **Que:** Facilita la información de reservas entre residentes y su comunicación a la hora de interactuar con los espacios comunes.

**f.** **A diferencia de:** Los métodos tradicionales basados en papel o soluciones digitales genéricas no diseñadas específicamente para la administración residencial, VICTUS Residencias ofrece una plataforma centralizada, intuitiva y en tiempo real que:

* **Elimina errores humanos** mediante automatización de reservas y seguimiento de cupos.
* **Facilita acceso inmediato a la información** para residentes y administradores, sin trámites físicos o demoras.
* **Integra comunicación directa** entre usuarios y administración, evitando conflictos por malentendidos o falta de claridad.
* **Se adapta exclusivamente** a las necesidades de conjuntos residenciales, con funciones como control de aforo, historial de reservas y notificaciones personalizadas.

**3**. **Visión box**

****

**4.** **Qué sí – Qué no:**

**Qué sí:** El proyecto incluirá funcionalidades como:

* Creación, modificación, eliminación y consulta de personas como Administradores, Residentes y Porteros que serán los usuarios del proyecto.
* Gestión de conjuntos residenciales, zonas comunes y Agendas específicas para cada zona común.
* Sistema de distribución de turnos en intervalos con un tiempo específico según el tiempo de inicio y final para cada zona común.
* Sistema de cambios en tiempo real para mostrar si un turno ya esta reservado o no para evitar solapamiento de reservas.
* Notificaciones automáticas para los residentes con el propósito de informar el estado de la reserva realizada en tiempo real.
* Sistema de confirmación de reservas como una asistencia del Residente a la zona común que va a usar siendo el usuario del Portero con el fin de visualizar que reservas hay en la zona común específica como una programación y dar Check o no Check a la reserva para así sancionar el residente que reservo y no asistió.
* Sistema de reportes con el fin de saber que tanto están reservando una zona común, las reservas que ha realizado un residente en cierto tiempo, cantidad de invitado que lleva un residente y datos de satisfacción de la app por los residentes para toma de decisiones administrativas.

**Qué no:** No se desarrollarán funcionalidades relacionadas con:

* La gestión de pagos o finanzas.
* Integración con sistemas externos como cámaras de seguridad.
* La automatización de servicios como limpieza o mantenimiento.
* Un portal de comunicación entre los residentes y la administración.

**En duda:**

* La implementación de una app móvil nativa está en discusión, dependiendo de la disponibilidad de recursos y el tiempo de desarrollo.
* El desarrollo de la interfaz de usuario (UI).
* La implementación de notificaciones en tiempo real.

**Seguros de hacer:** La creación de un portal web que gestione las reservas de espacio, así como la base de datos que soporte toda la información de los usuarios y reservas.

**5. La comunidad:**

**Roles Principales:**

**Administradores del conjunto:** Usuarios clave que gestionan las reservas y reciben reportes.

**Residentes:** Beneficiarios directos que utilizarán la aplicación para hacer reservas y recibir notificaciones.

**Sectores u Organizaciones:** Sectores Residenciales

**Equipo de Desarrollo:** Compuesto por los cuatro estudiantes, quienes diseñan y desarrollan el sistema.

**Proveedor de Hosting:** Para el alojamiento del sistema.

**Beneficiados:**

**Residentes:** Ahorro de tiempo y facilidad para hacer reservas.

**Administradores:** Mejora en la gestión de espacios y comunicación eficiente con los residentes.

**Comprometidos:**

**Equipo de Desarrollo:** Enfocados en la implementación y éxito del proyecto.

**Administración del conjunto:** Comprometidos con la adopción y uso del sistema.

**6. La solución:**

La solución consiste en desarrollar un sistema web que facilite la gestión de reservas de espacios comunes en conjuntos residenciales. La aplicación incluirá un calendario interactivo, un sistema de notificaciones para mantener a los residentes informados, y un panel de control para la administración que permita visualizar reportes y gestionar las solicitudes. La tecnología incluye un framework como INTELLIJ para desarrollar el BACKEND usando el lenguaje de programación JAVA, se usará de base de datos PostgreSQL y de FRONTEND usaremos el framework REACT para componetizar bien nuestra solución, los servicios que vamos a usar para publicar nuestra aplicación son RENDER.COM para subir el backend y la base de datos y para nuestro frontend usaremos NETLIFY.COM por temas de mejor rutamiento entre las paginas.

Para los servicio usaremos las versiones gratuitas para hacer pruebas y presentar la aplicación.

**7. Los miedos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgos** | **Descripción** | **Probabilidad de ocurrencia** | **Impacto** | **Plan de mitigación** |
| **Falta de experiencia en ciertas tecnologías** | La falta de conocimiento y experiencia en ciertos procesos que son necesarios para el desarrollo de las herramientas para la aplicación que podría dificultar el desarrollo de esta. | **40%** | **70%** | **Implementación de un plan de estudios individual o grupal para explorar las tecnologías necesarias para el desarrollo adecuado de las funcionalidades** |
| **Retrasos en la entrega** | La carga académica junto al trabajo particular de los integrantes del equipo de trabajo puede demorar los momentos de trabajo del equipo al momento de desarrollar. | **20%** | **80%** | **Manejo de 4 contextos los cuales son dependientes ir entregando lo que en el tiempo sea posible ya que no medimos con claridad la capacidad de cada integrante.** |
|  |  |  |  |  |

**Falta de experiencia en ciertas tecnologías:** Podría afectar la eficiencia del desarrollo.

**Retrasos en la entrega:** Dificultad para cumplir con los plazos establecidos debido a la carga académica o imprevistos técnicos.

**Problemas de seguridad:** Vulnerabilidades en el sistema que podrían comprometer la información de los usuarios.

**Impacto:** Si estos riesgos se materializan, pueden resultar en un proyecto incompleto o con errores significativos, afectando la experiencia del usuario y la adopción del sistema.

**Mitigación:** Capacitarse en tecnologías necesarias, definir un cronograma realista y realizar pruebas exhaustivas de seguridad.

**8. Tamaño:**

El proyecto tiene un tamaño mediano, con un tiempo estimado de ejecución de 4 a 6 meses, incluyendo etapas de planificación, desarrollo, pruebas y ajustes. Dado que somos cuatro estudiantes, cada uno podría tomar un rol específico para dividir las tareas y trabajar en paralelo, lo que podría optimizar el tiempo de desarrollo.

**9. El TRADE-OFF (Prioridades):**

**Atributos de calidad priorizados:**

**Usabilidad:** Alta prioridad para asegurar que la aplicación sea intuitiva y fácil de usar para todos los usuarios.

**Seguridad:** Alta prioridad para proteger los datos de los residentes y la integridad del sistema.

**Escalabilidad:** Media prioridad para permitir el crecimiento del sistema con nuevas funcionalidades en el futuro.

**Performance:** Media prioridad, ya que la velocidad de respuesta debe ser adecuada, pero no es crítico si es ligeramente más lento debido a la carga.

**10. ¿Cuánto cuesta?**

**Costos principales:**

**Desarrollo:** Principalmente el tiempo y esfuerzo de los cuatro estudiantes, que podría estimarse como costo de oportunidad.

**Hosting y Dominio:** Gastos asociados con el servidor web y el nombre de dominio pero solo usaremos las versiones gratuidad de RENDER Y NETLIFY para hacer pruebas al inicio.

**Software y Licencias:** Dependiendo de las herramientas utilizadas (por ejemplo, software de diseño o IDEs de pago), podría haber costos adicionales pero usaremos las versiones gratuitas por el momento.

**Testing y Deploy:** Potencialmente se pueden incluir algunos gastos en servicios de testing y despliegue, aunque podrían ser mínimos o evitados si se utilizan servicios gratuitos o de código abierto.

**VICTUS RESIDENCIAS**

1. **Requisitos Funcionales (Lo que el sistema debe hacer)**

**Usaremos como identificador único UUID para cada entidad y poderla diferenciar en la base de datos.**

**Gestión de Conjuntos Residenciales**

1. El sistema debe permitir crear, editar y eliminar conjuntos residenciales.
2. Cada conjunto debe tener un nombre único y un número de teléfono único.

**Gestión de Zonas Comunes**

1. El sistema debe permitir agregar zonas comunes a un conjunto residencial.
2. Cada zona común debe estar asociada a un conjunto existente.
3. Debe ser posible editar y eliminar zonas comunes de un conjunto.

**Gestión de Viviendas**

1. El sistema debe permitir registrar viviendas dentro de un conjunto (casa, apartamento, glamping, etc).
2. Debe ser posible editar y eliminar viviendas dentro de un conjunto.

**Gestión de Residentes**

1. El sistema debe permitir registrar residentes con sus datos personales.
2. Un residente debe estar asociado a una vivienda específica.
3. Un residente no puede estar asociado a más de una vivienda al mismo tiempo.
4. Debe permitir la edición y eliminación de residentes.

**Gestión de la Agenda de Reservas**

1. El administrador debe poder definir los turnos y horarios permitidos en cada zona común.
2. No deben existir horarios sobrepuestos dentro de una misma zona común.
3. El sistema debe mostrar los turnos disponibles para cada zona común.

**Sistema de Reservas**

1. Los residentes deben poder reservar turnos en las zonas comunes disponibles.
2. Una vez realizada una reserva, el turno debe bloquearse para que no pueda ser reservado nuevamente.
3. Un residente solo puede tener una reserva activa en una misma zona común a la vez.
4. El sistema debe permitir la cancelación de reservas por parte del residente o del administrador.
5. Se debe notificar a los residentes cuando su reserva sea confirmada o cancelada.
6. El sistema debe permitir la consulta del historial de reservas.

**2. Requisitos No Funcionales (Cómo debe funcionar el sistema)**

**Rendimiento y Escalabilidad**

1. El sistema debe responder en menos de 3 segundos para consultas y reservas.
2. Debe ser escalable para admitir múltiples conjuntos residenciales sin afectar el rendimiento.

**Usabilidad**

1. La interfaz debe ser intuitiva y fácil de usar para administradores y residentes.
2. Debe estar optimizada para dispositivos móviles y de escritorio.

**Seguridad**

1. El acceso al sistema debe ser mediante autenticación con usuario y contraseña.
2. Los administradores y residentes deben tener diferentes niveles de permisos.
3. Los datos personales de los residentes deben ser protegidos según normativas de privacidad.

**Disponibilidad y Fiabilidad**

1. El sistema debe estar disponible al menos el 99% del tiempo.
2. Debe contar con copias de seguridad automáticas de la base de datos.

**3. Requisitos de Información (Datos que manejará el sistema)**

**Información de Conjuntos Residenciales**

* Nombre del conjunto residencial (único).
* Teléfono de contacto (único).
* Dirección
* Descripción
* Administrador

**Información de Zonas Comunes**

* Nombre de la zona común (piscina, gimnasio, etc.).
* Descripción de la zona.
* Conjunto residencial al que pertenece.
* Imagen
* Capacidad
* Tiempo de uso por día
* Normas

**Información de Viviendas**

* Tipo de vivienda (casa, apartamento, glamping, etc).
* Numero de vivienda.
* Tipo de zona de vivienda (Torres, Manzana , Lago, etc).
* Numero de zona de vivienda.
* Conjunto residencial al que pertenece.

**Información de Residentes**

* Nombre completo.
* Tipo de identificación.
* Número de identificación.
* Fecha de nacimiento.
* Contacto (correo, teléfono).
* Vivienda asignada.
* Contraseña.

**Información de Reservas**

* Zona común reservada.
* Residente que hizo la reserva.
* Fecha y hora de la reserva.
* Estado de la reserva (activa, cancelada, completada).
* Numero de invitados.

**Información de Agenda de Reservas**

* Horarios y turnos disponibles.
* Tipo de recurrencia.
* Fecha y hora de inicio o fin.
* Administrador que los configuró.

Sprints para separar producción

**Fase 1: Base del sistema**

* PBIs: 1-3 (Conjuntos residenciales), 4-6 (Zonas comunes), 7-9 (Viviendas).

**Fase 2: Gestión de residentes y reservas iniciales**

* PBIs: 10-12 (Residentes), 13-15 (Agenda de reservas

**Fase 3: Reservas y mejoras finales**

* PBIs: 16-20 (Sistema de reservas).

Link Azure DevOps : <https://dev.azure.com/ViktusResidencias/Victus%20Residencias>

PBIS

**1. Gestión de Conjuntos Residenciales**

1. **Crear entidad Conjunto Residencial con atributos (nombre, teléfono único, dirección ,descripción y administrador).**
2. **Interfaz para gestionar conjuntos residenciales (crear, editar, eliminar).**
3. **Validaciones para evitar conjuntos con el mismo nombre o teléfono.**

**2. Gestión de Zonas Comunes**

1. **Crear entidad Zona Común con atributos (nombre, descripción, conjunto asociado, imagen, capacidad, tiempo de uso por día y normas).**
2. **Interfaz para gestionar zonas comunes (crear, editar, eliminar).**
3. **Restricción para asociar zonas solo a conjuntos existentes.**

**3. Gestión de Viviendas**

1. **Crear entidad Vivienda con atributos (tipo: casa, apartamento, glamping, conjunto asociado, tipo de zona de vivienda , numero de vivienda , numero de zona de vivienda).**
2. **Interfaz para gestionar viviendas (crear, editar, eliminar).**
3. **Validación para evitar viviendas duplicadas en un conjunto.**

**4. Gestión de Residentes**

1. **Crear entidad Residente con atributos (nombre, identificación, contacto, vivienda asignada, tipo de documento , fecha de nacimiento y contraseña).**
2. **Interfaz para gestionar residentes (crear, editar, eliminar, asignar vivienda).**
3. **Restricción para no asignar múltiples residentes a la misma vivienda si aplica.**

**5. Gestión de Agenda de Reservas**

1. **Crear entidad Agenda de Reservas con atributos (zona común, horario permitido, administrador que define horarios).**
2. **Interfaz para gestionar la agenda (crear horarios, modificar turnos disponibles).**
3. **Restricción para evitar solapamiento de horarios en una misma zona común.**

**6. Sistema de Reservas**

1. **Crear entidad Reserva con atributos (residente, zona común, fecha y hora reservada).**
2. **Interfaz para que residentes reserven turnos (mostrar solo los disponibles).**
3. **Bloqueo automático de turnos ya reservados.**
4. **Notificación al residente al confirmar su reserva (correo o mensaje interno).**
5. **Interfaz para visualizar reservas activas y pasadas.**

**HISTORIAS DE USUARIO**

**1. Gestión de Conjuntos Residenciales**

**HU01 - Crear conjunto residencial**

Como administrador del sistema Quiero poder registrar un conjunto residencial con su nombre y teléfono único Para gestionar las reservas de sus zonas comunes.

**HU02 - Editar conjunto residencial**

Como administrador del sistema Quiero poder modificar los datos de un conjunto residencial Para actualizar información relevante cuando sea necesario.

**HU03 - Eliminar conjunto residencial**

Como administrador del sistema Quiero poder eliminar un conjunto residencial si ya no se usa Para mantener la base de datos limpia y actualizada.

**2. Gestión de Zonas Comunes**

**HU04 - Registrar una zona común**

Como administrador del sistema Quiero poder agregar zonas comunes a un conjunto residencial Para que los residentes puedan reservarlas.

**HU05 - Editar una zona común**

Como administrador del sistema Quiero modificar los datos de una zona común Para actualizar su disponibilidad o características.

**HU06 - Eliminar una zona común**

Como administrador del sistema Quiero poder eliminar una zona común Para gestionar correctamente los espacios disponibles.

**3. Gestión de Viviendas**

**HU07 - Registrar una vivienda**

Como administrador del sistema Quiero agregar viviendas dentro de un conjunto residencial Para asignarlas a los residentes correspondientes.

**HU08 - Editar una vivienda**

Como administrador del sistema Quiero modificar los datos de una vivienda Para corregir información o actualizar su estado.

**HU09 - Eliminar una vivienda**

Como administrador del sistema Quiero poder eliminar una vivienda Para mantener el registro actualizado.

**4. Gestión de Residentes**

**HU10 - Registrar un residente**

Como administrador del sistema Quiero poder registrar nuevos residentes Para que puedan hacer reservas en las zonas comunes.

**HU11 - Asignar una vivienda a un residente**

Como administrador del sistema Quiero asignar una vivienda a un residente Para identificar en qué lugar vive dentro del conjunto.

**HU12 - Editar información de un residente**

Como administrador del sistema Quiero poder modificar los datos de un residente Para actualizar su información en caso de cambios.

**HU13 - Eliminar un residente**

Como administrador del sistema Quiero poder eliminar un residente Para mantener actualizado el registro de personas activa.

**5. Gestión de la Agenda de Reservas**

**HU14 - Configurar turnos y horarios de reservas**

Como administrador del sistema Quiero definir los horarios en los que se pueden reservar las zonas comunes Para organizar correctamente la disponibilidad de los espacios.

**HU15 - Evitar solapamiento de horarios**

Como administrador del sistema Quiero que el sistema evite horarios superpuestos en la misma zona común Para evitar conflictos entre los residentes.

**HU16 - Visualizar horarios disponibles**

Como residente del conjunto Quiero ver los horarios disponibles de las zonas comunes Para seleccionar uno que se ajuste a mi necesidad.

**6. Sistema de Reservas**

**HU17 - Realizar una reserva**

Como residente del conjunto Quiero poder reservar un turno en una zona común Para asegurarme de que estará disponible para mi uso.

**HU18 - Bloquear un turno reservado**

Como sistema Quiero bloquear automáticamente un turno reservado Para evitar que otro residente lo tome.

**HU19 - Cancelar una reserva**

Como residente del conjunto Quiero poder cancelar una reserva que ya no necesito Para liberar el turno para otra persona.

**HU20 - Notificar confirmación de reserva**

Como residente del conjunto Quiero recibir una notificación cuando mi reserva sea confirmada Para estar seguro de que el turno es mío.

**HU21 - Consultar historial de reservas**

Como residente del conjunto Quiero poder ver mis reservas pasadas y futuras Para llevar un control de mis usos de las zonas comunes.