**Documento del Diagrama de Base de Datos**

***Sistema Unidad Territorial***

***Fecha: 16/10/2024***

Contenido

[1. Introducción 4](#_Toc180099851)

[2. Diagrama de la Base de Datos 5](#_Toc180099852)

[3. Descripción de Tablas 6](#_Toc180099853)

[Tabla: usuarios 6](#_Toc180099854)

[Tabla: actividades 7](#_Toc180099855)

[Tabla: participantes 7](#_Toc180099856)

[Tabla: espaciosPublicos 8](#_Toc180099857)

[Tabla: fechasReservadas 8](#_Toc180099858)

[Tabla: noticias 9](#_Toc180099859)

[Tabla: proyectos 9](#_Toc180099860)

[Tabla: documentosRequeridos 10](#_Toc180099861)

[Tabla: postulantes 10](#_Toc180099862)

[Tabla: documentosSubidos 10](#_Toc180099863)

[Relaciones 11](#_Toc180099864)

[4. Base de datos no relacional con Firebase 12](#_Toc180099865)

[Características clave: 12](#_Toc180099866)

[Uso de Firebase en el sistema: 12](#_Toc180099867)

[5. Conclusión 12](#_Toc180099868)

**Historial de Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
|  | **1.0** | **Alejandro villa** | **Departamento de Informática** | **Versión inicial del documento** |

**Información del Proyecto**

|  |  |
| --- | --- |
| Empresa / Organización | Departamento de Informática y Telecomunicaciones de Duoc UC. |
| Proyecto | Sistema Unidad Territorial |
| Fecha de preparación | 29/08/2024 |
| Cliente | Municipalidad de San Bernardo |
| Patrocinador principal | Escuela de ingeniería informática DuocUC |
| Líder de Proyecto | Alejandro Villa |
| Gerente / Líder de Análisis de negocio y requerimientos |  |

**Aprobaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| **Alejandro Villa** | **Líder de proyecto** | **Escuela de ingeniería** | **16-10-2024** |  |

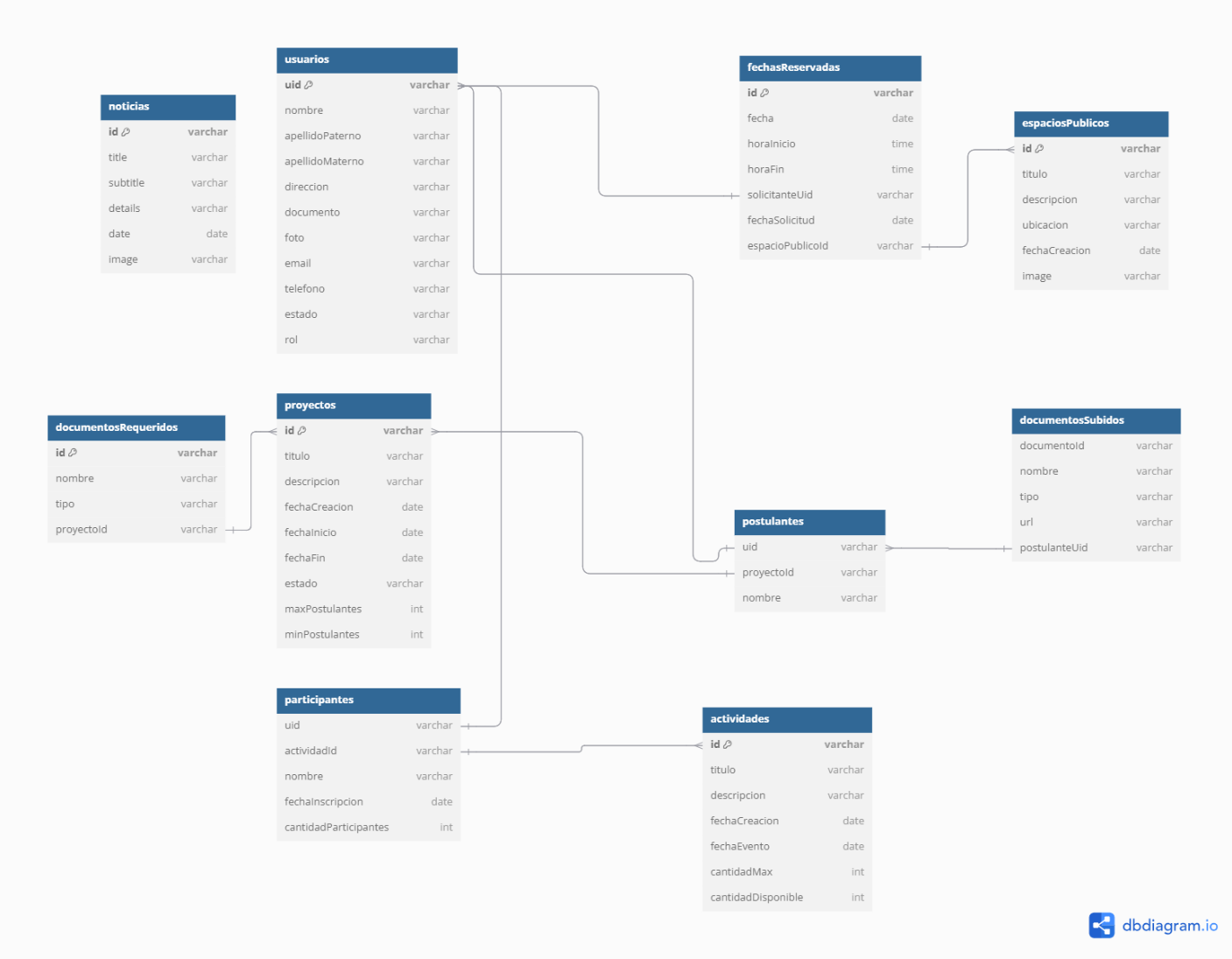
# 1. Introducción

El proyecto "Sistema Unidad Territorial" tiene como objetivo desarrollar una solución tecnológica que mejore la gestión de las juntas de vecinos, enfocándose en la organización de actividades, manejo de recursos y comunicaciones. Ante los desafíos que enfrentan estas organizaciones, especialmente la obsolescencia de herramientas tradicionales, el sistema propuesto busca optimizar tanto la eficiencia interna como la experiencia del usuario final.

Este documento presenta el diagrama de base de datos del sistema, que es fundamental para implementar funcionalidades clave como la inscripción y gestión de vecinos, la emisión de certificados de residencia y la automatización de solicitudes. La estructura de la base de datos incluye tablas esenciales como usuarios, actividades, participantes, espaciosPublicos, fechasReservadas, noticias, proyectos, documentosRequeridos, postulantes, y documentosSubidos. Cada tabla está diseñada para gestionar información específica y facilitar la interconexión entre diferentes componentes del sistema, garantizando una gestión eficaz de la información.

La base de datos no solo respalda la operativa diaria del sistema, sino que también es clave para la toma de decisiones informadas, la generación de reportes y el análisis de datos. Así, contribuye significativamente al objetivo general de mejorar la gestión de las unidades territoriales de las juntas de vecinos mediante una solución accesible y tecnológica que cubra todas sus necesidades operativas y de comunicación.

# 2. Diagrama de la Base de Datos

****

# 3. Descripción de Tablas

Esta sección detalla las tablas que componen la base de datos del "Sistema Unidad Territorial", incluyendo sus campos, tipos de datos y relaciones entre ellas. Cabe mencionar que la base de datos es no relacional y se está utilizando con Firebase, pero el diagrama se ha estructurado de esta manera para facilitar la comprensión de las interacciones y dependencias entre los distintos elementos del sistema.

## Tabla: usuarios

Descripción: Almacena la información de los usuarios registrados en el sistema, incluyendo sus datos personales y de contacto.  
Campos:

uid (varchar, pk): Identificador único del usuario.

nombre (varchar): Nombre del usuario.

apellidoPaterno (varchar): Apellido paterno del usuario.

apellidoMaterno (varchar): Apellido materno del usuario.

direccion (varchar): Dirección de residencia del usuario.

documento (varchar): Número de documento de identificación.

foto (varchar): URL de la foto del usuario.

email (varchar): Correo electrónico del usuario.

telefono (varchar): Número de teléfono del usuario.

estado (varchar): Estado de la cuenta (activo, inactivo, etc.).

rol (varchar): Rol del usuario en el sistema (ej. administrador, vecino).

## Tabla: actividades

**Descripción:** Almacena información sobre las actividades organizadas por la junta de vecinos.  
**Campos:**

id (varchar, pk): Identificador único de la actividad.

titulo (varchar): Título de la actividad.

descripcion (varchar): Descripción de la actividad.

fechaCreacion (date): Fecha de creación de la actividad.

fechaEvento (date): Fecha del evento.

cantidadMax (int): Cantidad máxima de participantes permitidos.

cantidadDisponible (int): Cantidad de cupos disponibles.

## Tabla: participantes

**Descripción:** Registra a los usuarios que participan en las actividades.  
**Campos:**

uid (varchar, fk): Identificador único del usuario (relación con usuarios).

actividadId (varchar, fk): Identificador de la actividad (relación con actividades).

nombre (varchar): Nombre del participante.

fechaInscripcion (date): Fecha de inscripción a la actividad.

cantidadParticipantes (int): Cantidad de participantes por usuario en la actividad.

## Tabla: espaciosPublicos

Descripción: Almacena información sobre los espacios públicos disponibles para ser reservados por los vecinos.  
Campos:

id (varchar, pk): Identificador único del espacio.

titulo (varchar): Título del espacio.

descripcion (varchar): Descripción del espacio.

ubicacion (varchar): Ubicación del espacio.

fechaCreacion (date): Fecha de creación del registro del espacio.

image (varchar): URL de la imagen del espacio.

## Tabla: fechasReservadas

Descripción: Registra las reservas realizadas por los usuarios para utilizar espacios públicos.  
Campos:

id (varchar, pk): Identificador único de la reserva.

fecha (date): Fecha de la reserva.

horaInicio (time): Hora de inicio de la reserva.

horaFin (time): Hora de fin de la reserva.

solicitanteUid (varchar, fk): Identificador del solicitante (relación con usuarios).

fechaSolicitud (date): Fecha en que se realizó la solicitud.

espacioPublicoId (varchar, fk): Identificador del espacio público (relación con espaciosPublicos).

## Tabla: noticias

**Descripción:** Almacena noticias relevantes para la comunidad.  
**Campos:**

id (varchar, pk): Identificador único de la noticia.

title (varchar): Título de la noticia.

subtitle (varchar): Subtítulo de la noticia.

details (varchar): Detalles de la noticia.

date (date): Fecha de publicación de la noticia.

image (varchar): URL de la imagen asociada a la noticia.

## Tabla: proyectos

**Descripción:** Registra proyectos propuestos por los vecinos.  
**Campos:**

id (varchar, pk): Identificador único del proyecto.

titulo (varchar): Título del proyecto.

descripcion (varchar): Descripción del proyecto.

fechaCreacion (date): Fecha de creación del proyecto.

fechaInicio (date): Fecha de inicio del proyecto.

fechaFin (date): Fecha de finalización del proyecto.

estado (varchar): Estado del proyecto (activo, finalizado, etc.).

maxPostulantes (int): Número máximo de postulantes permitidos.

minPostulantes (int): Número mínimo de postulantes requeridos.

## Tabla: documentosRequeridos

Descripción: Almacena los documentos requeridos para la postulación a proyectos.  
Campos:

id (varchar, pk): Identificador único del documento requerido.

nombre (varchar): Nombre del documento.

tipo (varchar): Tipo de documento (ej. identificación, propuesta).

proyectoId (varchar, fk): Identificador del proyecto asociado (relación con proyectos).

## Tabla: postulantes

**Descripción:** Registra a los usuarios que postulan a proyectos.  
**Campos:**

uid (varchar, fk): Identificador del postulante (relación con usuarios).

proyectoId (varchar, fk): Identificador del proyecto al que se postula (relación con proyectos).

nombre (varchar): Nombre del postulante.

## Tabla: documentosSubidos

**Descripción:** Almacena los documentos subidos por los postulantes.  
**Campos:**

documentoId (varchar, fk): Identificador del documento (relación con documentosRequeridos).

nombre (varchar): Nombre del documento subido.

tipo (varchar): Tipo de documento (ej. PDF, imagen).

url (varchar): URL del documento en la nube.

postulanteUid (varchar, fk): Identificador del postulante (relación con postulantes).

## Relaciones

actividades.id > participantes.actividadId: Relación entre actividades y participantes.

usuarios.uid > participantes.uid: Relación entre usuarios y participantes.

usuarios.uid > fechasReservadas.solicitanteUid: Relación entre reservas y usuarios.

espaciosPublicos.id > fechasReservadas.espacioPublicoId: Relación entre espacios públicos y fechas reservadas.

proyectos.id > documentosRequeridos.proyectoId: Relación entre proyectos y documentos requeridos.

proyectos.id > postulantes.proyectoId: Relación entre proyectos y postulantes.

usuarios.uid > postulantes.uid: Relación entre usuarios y postulantes.

postulantes.uid > documentosSubidos.postulanteUid: Relación entre postulantes y documentos subidos.

documentosRequeridos.id > documentosSubidos.documentoId: Relación entre documentos requeridos y documentos subidos.

# 4. Base de datos no relacional con Firebase

La implementación de una base de datos no relacional utilizando Firebase se centra en la flexibilidad y escalabilidad que ofrece este servicio. Al adoptar Firebase, se busca optimizar el manejo de la información, permitiendo una interacción más dinámica y en tiempo real con los datos.

## Características clave:

Almacenamiento en tiempo real: Firebase permite que los datos se sincronicen en tiempo real, lo que facilita una experiencia fluida para los usuarios al interactuar con el sistema.

Escalabilidad: La base de datos puede crecer de manera eficiente a medida que aumentan los usuarios y las necesidades del sistema, sin comprometer el rendimiento.

Seguridad: Firebase proporciona reglas de seguridad configurables, asegurando que solo los usuarios autorizados tengan acceso a ciertos datos, lo cual es esencial para proteger la información sensible de los usuarios y de la comunidad.

## Uso de Firebase en el sistema:

Se utilizará para gestionar la autenticación de usuarios, permitiendo un acceso seguro y controlado a las distintas funcionalidades del sistema.

El almacenamiento de datos sobre usuarios, actividades, y documentos se realizará en Firebase, utilizando estructuras de colecciones y documentos que reflejan la información de las tablas anteriormente mencionadas.

Las interacciones del sistema, como la creación y gestión de actividades, reservas de espacios públicos, y postulaciones a proyectos, se llevarán a cabo a través de Firebase, aprovechando sus capacidades en tiempo real para mantener a todos los usuarios actualizados.

# 5. Conclusión

Este documento proporciona una visión integral del diagrama de la base de datos para el "Sistema Unidad Territorial", destacando la interconexión entre diferentes entidades y la flexibilidad que ofrece el uso de Firebase como base de datos no relacional. A través de esta estructura, se busca asegurar que el sistema cumpla con los requerimientos funcionales y no funcionales, optimizando la experiencia de los usuarios y facilitando la gestión de las juntas de vecinos.